

UNIVERSIDADE ESTADUAL PAULISTA

“JÚLIO DE MESQUISTA FILHO”

Faculdade de Ciências – Câmpus Bauru

Bacharelado em Ciência da Computação

DANIELLE DOMENEGHETTI CREPALDI

**DESENVOLVIMENTO DE OBJETO DE APRENDIZAGEM PARA
DISPOSITIVOS MÓVEIS EXPLORANDO OS SENSORES INTERNOS
DO SISTEMA OPERACIONAL ANDROID**

BAURU

2014

DANIELLE DOMENEGHETTI CEPALDI

**DESENVOLVIMENTO DE OBJETO DE APRENDIZAGEM PARA
DISPOSITIVOS MÓVEIS EXPLORANDO OS SENSORES INTERNOS
DO SISTEMA OPERACIONAL ANDROID**

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Computação da Universidade Estadual
Júlio de Mesquita Filho – UNESP, Câmpus
de Bauru.

Orientador: Prof. Dr. Wilson Massashiro
Yonezawa

BAURU

2014

AGRADECIMENTOS

Agradeço a atenção e dedicação do meu orientador, Prof. Dr. Wilson Massashiro Yonezawa, que esteve disponível durante todo o desenvolvimento do projeto para auxiliar e esclarecer dúvidas pertinentes à criação do objeto de aprendizagem.

Agradeço, também, à aluna de Designer, Daniela Yumi Onari, pela participação e ajuda prestada para definir as funcionalidades e interface do sistema.

RESUMO

O modelo tradicional de ensino pouco se modernizou nas últimas décadas. Ao entrar em contato com esse sistema, é normal que os novos alunos se sintam desmotivados em realizar as atividades propostas em sala de aula. Segundo Prensky (2001) esses alunos, denominados de Nativos Digitais, nascem com uma nova maneira de pensar e aprender e se sentem estimulados e motivados com atividades que os convidem a interagir. Para isso, foi proposto o desenvolvimento de um objeto de aprendizagem na forma de aplicativo para *tablets* Android, explorando os sensores internos disponíveis nos mesmos, com a finalidade de oferecer uma atividade interativa ao aluno sobre os conceitos físicos envolvidos no processo da fotossíntese das plantas. Para a construção do mesmo foram empregados textos informativos, escritos em linguagem didática e de fácil compreensão, imagens ilustrativas e animações. Além disso, utilizou-se o sensor de luminosidade na atividade interativa referente ao processo da fotossíntese para que o aluno pudesse observar e compreender como o ambiente é capaz de interferir neste processo.

Palavras-chave: Objetos de Aprendizagem. Tecnologia na educação. Android. *Tablets*. Sensores internos dos dispositivos Android.

ABSTRACT

The traditional model of teaching little is modernized in recent decades. When in contact with this system, it is normal that the new-generation students feel unmotivated in carrying out the proposed activities in the classroom. Considers Prensky (2001) these students, called Digital Natives, born with a modern way of thinking and learn and feel encouraged and motivated with activities that invite to interact. For this reason, it has been proposed the development of an object of learning in the form of application for Android tablets, exploring the internal sensors available in them, with the purpose of offering an interactive activity to students on the physical concepts involved in the process of photosynthesis in plants. For the construction, informational texts written in didactic language and easy to understand, illustrative images and animations were employed. In addition, we used the light sensor in the interactive activity on the process of photosynthesis so that the student could observe and understand how the environment is able to interfere with this process.

Keywords: Learning Objects. Technology in education. Android. *Tablets*. Android device sensors.

LISTA DE FIGURAS

Figura 1 - Etapas do desenvolvimento do modelo ADDIE	19
Figura 2 - Ciclo de Vida do Processo de Desenvolvimento de Software (usando o Modelo Cascata)	20
Figura 3 - Figura ilustrativa do objeto Trigonometria na ponte	22
Figura 4 - Figura ilustrativa do objeto Futebol no país da matemática	23
Figura 5 - Figura ilustrativa do objeto Aprendendo as Leis de Newton.	23
Figura 6 - Figura ilustrativa do objeto A química dentro de um bolo	24
Figura 7 - Figura ilustrativa do objeto Ondas de Rádio	24
Figura 8 - Exemplos de Objetos de Aprendizagem para dispositivos móveis	25
Figura 9 - Exemplo de um objeto de aprendizagem sobre Biologia, desenvolvido pela EvoBooks	25
Figura 10 - Atual adoção das versões Android presentes nos dispositivos móveis ..	27
Figura 11 - Arquitetura da plataforma Android	28
Figura 12 - Segmento dos sistemas operacionais para <i>smartphones</i> - participação de mercado no final do terceiro trimestre de 2013.	30
Figura 13 - Segmento dos sistemas operacionais para <i>tablets</i> - participação de mercado no final do terceiro trimestre de 2013.	30
Figura 14 - Esquema de funcionamento de um sensor.....	33
Figura 15 - Unidade de iluminação lux (lx).	43
Figura 16- Funcionamento do sensor de proximidade.	44
Figura 17 - Sistema de coordenadas Android.	46
Figura 18 - Regra da mão direita.....	49
Figura 19 - Princípio da Lei de Hooke.	50
Figura 20 - Sistema de massa usado para medir aceleração	50
Figura 21 - Força aplicada a uma massa suspensa por molas (g é a unidade de aceleração e equivale a 9.8 m/s^2)	51
Figura 22 - Representação de um giroscópio.....	52
Figura 23 - Declinação magnética a partir da direção do Norte Verdadeiro	54
Figura 24 - Os eixos referentes a <code>getOrientation()</code>	56
Figura 25 - Representação dos valores retornados pelo método <code>getOrientation()</code>	57
Figura 26 - Exatidão versus precisão	61

Figura 27 - Implementação do filtro passa baixa	64
Figura 28 - Implementação do filtro passa baixa inverso	65
Figura 29 - Implementação de um filtro passa-alta	66
Figura 30 - Desenho esquemático de cloroplasto visto em corte	70
Figura 31 - Etapas da fotossíntese.....	70
Figura 32 - Esquema de fotossíntese nos cloroplastos.	73
Figura 33 - Fases de desenvolvimento do projeto de TCC.	74
Figura 34 - Diagrama das telas do aplicativo.	76
Figura 35 - Imagens das telas provisórias do projeto – Tela da Linha do tempo e Aparelho fotossintético.	77
Figura 36 – Representação de algumas telas do protótipo do aplicativo - Tela da Linha do tempo, Aparelho fotossintético e Telas do Processo da fotossíntese (à esquerda, etapa fotoquímica e à direita, etapa química).....	77
Figura 37 - Ciclo de vida de uma <i>activity</i>	79
Figura 38 - Exemplo da utilização dos <i>fragments</i> na construção de <i>layouts</i> para <i>tablets</i> e <i>smartphones</i>	80
Figura 39 - Exemplo de aplicativo utilizando o Navigation Drawer	81
Figura 40 - Exemplo de utilização do ViewPager.	81
Figura 41 - Exemplo de utilização do ListView.	82
Figura 42 - Representação do uso do arquivo <i>string.xml</i>	85
Figura 43 - Trecho de código, extraído do projeto, responsável por obter os dados do tipo String do arquivo <i>string.xml</i>	85
Figura 44 - Método onCreateView() da classe <i>ProcessoFotossinteseFragment</i>	86
Figura 45 - Trecho do código extraído da classe <i>ProcessoFotossinteseFragment</i>	87
Figura 46 - Implementação do método onSensorChanged() na classe <i>ProcessoFotossinteseFragment</i>	88
Figura 47 - <i>Tablet</i> utilizado para testes do aplicativo	89

LISTA DE QUADROS

Quadro 1 - Recursos de Hardware do Android	32
Quadro 2 - Tipos de sensores suportados pela plataforma Android	35
Quadro 3 - Disponibilidade de sensores de acordo com a versão do Android	38
Quadro 4 - Valores retornados pelos sensores disponíveis na plataforma Android .	57
Quadro 5 - Etapas do processo da fotossíntese demonstradas no aplicativo.	84

SUMÁRIO

1. INTRODUÇÃO	11
1.1 Tema e problema de pesquisa	11
1.2 Relevância do tema proposto	12
1.3 Objetivos da pesquisa	13
1.3.1 Objetivo Geral	13
1.3.2 Objetivos Específicos	13
1.4 Organização do trabalho	14
2. REFERENCIAL TEÓRICO	15
2.1 Objetos de Aprendizagem	15
2.1.1 Definição	16
2.1.2 Aspectos Pedagógicos	18
2.1.3 Modelo ADDIE para o desenvolvimento de Objetos de Aprendizagem	18
2.1.4 Objetos de Aprendizagem no Brasil	20
2.2 Sistema Operacional Android	26
2.2.1 Plataforma	26
2.2.2 Arquitetura do sistema operacional Android	27
2.2.3 Desenvolvimento de software para Android	29
2.3 Sensores internos disponíveis nos dispositivos Android	33
2.3.1 Sensores	33
2.3.2 Tipos de sensores internos disponíveis nos dispositivos Android	34
2.3.3 Categorias	37
2.3.4 Disponibilidade	38
2.3.5 Como utilizar os sensores	39
2.3.6 Descrição dos sensores internos	42

2.3.7	Interpretando os valores dos sensores.....	57
2.3.8	Erros e processamento dos valores vindos dos sensores internos	59
2.3.9	Filtros.....	63
2.4	Fotossíntese.....	66
2.4.1	Definição e Importância	67
2.4.2	Histórico.....	67
2.4.3	O processo da fotossíntese	70
2.4.4	A física na fotossíntese.....	72
2.4.5	A química na fotossíntese	73
3.	SOLUÇÃO PROPOSTA.....	74
3.1	Fase de Análise	74
3.2	Fase de Projeto.....	75
3.3	Fase de Desenvolvimento.....	78
3.4	Fases de Implementação e Validação	88
4.	CONCLUSÃO	90
4.1	Contribuições e trabalhos futuros.....	91
	REFERÊNCIAS BIBLIOGRÁFICAS	92
	APÊNDICES.....	99
	APÊNDICE A - Diagrama de classes do projeto.....	99

1. INTRODUÇÃO

1.1 Tema e problema de pesquisa

A cada dia que passa, a tecnologia vem se incorporando às atividades cotidianas e potencializando a capacidade humana. Com o advento da Internet, a sociedade passou a viver em um mundo conectado, seja pelo fato de que as informações estão disponíveis a qualquer momento, seja pelo fato de que a comunicação se tornou instantânea.

Se antes as informações estavam presentes somente nos veículos tradicionais de comunicação (rádios, televisão, jornais e revistas), hoje elas estão, literalmente, na palma das mãos. Com a evolução dos equipamentos eletrônicos, celulares, *smartphones* e *tablets* passaram a ser responsáveis por desempenhar grande parte das atividades que antes eram realizadas apenas por computadores, sendo o acesso à Internet a principal delas.

Segundo o Instituto Brasileiro de Opinião Pública e Estatística (IBOPE, 2013), no primeiro trimestre de 2013 o Brasil registrou 102,3 milhões de usuários conectados à Internet. Esses dados consideram pessoas de 16 anos ou mais de idade que possuem acesso em qualquer ambiente (domicílios, trabalho, escolas, *lan houses* e outros locais), além de crianças e adolescentes com idades que variam de 2 a 15 anos que têm acesso em casa.

Dados divulgados pela pesquisa ‘Geração Interativas Brasil – Crianças e Adolescentes Diante das Telas’, lançados no fim de 2012 pela Fundação Telefônica, em parceria com o Ibope, a Escola do Futuro/USP e a Universidade de Navarra, apontaram que 47% das crianças (de 6 a 9 anos de idade) e 75% dos jovens (10 a 18 anos) costumam navegar na rede, normalmente através de *smartphones* e celulares. (LIMA, 2013).

Como comprovam os dados estatísticos, as novas gerações já nascem conectadas. Segundo reportagem da Revista Veja (2001) a geração Z, como é chamada, é mais receptiva as novidades tecnológicas e tem certa facilidade para utilizá-las. O “Z”, que em inglês significa “fazer algo muito rapidamente” e também “energia” ou “entusiasmo”, vem de “zapear” e se remete à capacidade que esta geração possui de migrar rapidamente de um tipo de tecnologia para outro.

“Os alunos dessa nova geração, denominada de Nativos Digitais, são todos ‘falantes nativos’ da linguagem digital dos computadores, vídeo games e internet”. (PRENSKY, 2001, tradução nossa). Com uma nova maneira de pensar e aprender, essa geração desafia as instituições a saírem do modelo tradicional de ensino para uma nova forma de aprendizagem. Como tornar mais atraente o ensino e escola para esta nova geração?

1.2 Relevância do tema proposto

[...] a geração Z (ou geração digital) nasceu e cresceu com um computador à disposição e ganhou intimidade com um telefone celular muito cedo. Portanto, tem infiltrado na própria pele a percepção de mundo através da tecnologia, da estética, da velocidade. Não concebem mais o seu cotidiano sem estes acessórios (que já se tornaram roupa) não dá para viver sem eles. Não há no repertório destes jovens a experiência capaz de fazer de forma mais artesanal, manual, o que eles fazem com computadores e celulares. (CORTONI, 2006).

Ao analisar os impactos causados na educação por essas inovações tecnológicas, é possível perceber que estas influenciam a percepção da cultura e o mundo ao redor dos ‘nativos digitais’, bem como o modo de estudar, aprender e pesquisar. (SANTOS; SCARABOTTO; MATOS, 2011).

No entanto, o modelo tradicional de ensino pouco se modernizou nas últimas décadas. O quadro negro, as fileiras de alunos e a figura do professor à frente da sala de aula, presentes ainda na maioria das escolas, mantém o mesmo padrão de ensino do século XIX.

Grande parte das instituições continuam exigindo trabalhos feitos em papel, passando matéria na lousa e insistindo no ensino de cálculos manuais e fórmulas complicadas que poucos estudantes entendem.

Ao entrar em contato com esse método de ensino, é natural que a nova geração se sinta desmotivada. Para Prensky (2001), os ‘nativos digitais’ se sentem estimulados e motivados com atividades que os convidem a interagir. Querem ser ‘usuários’ e não simplesmente observadores.

De acordo com Perrenoud (2000, p. 125), “as crianças nascem em uma cultura em que se clica e o dever dos professores é inserir-se no universo de seus alunos”. Dessa maneira, as escolas precisam se adaptar a um novo modelo de ensino, focado no desenvolvimento de materiais didáticos e acadêmicos que se utilizem das

tecnologias digitais e atraíam a atenção do aluno. Andrade (2011, p. 15) afirma que “as vantagens de se utilizar as tecnologias como ferramenta pedagógica é estimular os alunos, dinamizar o conteúdo, e fomentar a autonomia e a criatividade”.

Visualizar, através de conteúdos interativos, conceitos como a ação da gravidade ou o que acontece com as partículas e moléculas quando há o aumento da temperatura, por exemplo, poderiam ser métodos eficazes de aplicar a tecnologia a favor do conhecimento em situações reais do cotidiano.

Para essa finalidade, faz-se necessário explorar novas formas de objetos de aprendizagem (OA). Os OAs podem ser definidos como:

[...] qualquer recurso, suplementar ao processo de aprendizagem, que pode ser reusado para apoiar a aprendizagem. O termo objeto educacional (learning object) geralmente aplica-se a materiais educacionais projetados e construídos em pequenos conjuntos com vistas a maximizar as situações de aprendizagem onde o recurso pode ser utilizado. (TAROUCO; FABRE; TAMUSIUNAS, 2003, p. 2)

1.3 Objetivos da pesquisa

1.3.1 Objetivo Geral

O objetivo deste projeto é desenvolver um objeto de aprendizagem em forma de aplicativo para *tablets*, explorando os sensores internos existentes nestes dispositivos, com a finalidade de facilitar o ensino-aprendizagem do processo de fotossíntese para os estudantes do Ensino Médio.

1.3.2 Objetivos Específicos

- a) Compreender o funcionamento dos sensores nos dispositivos móveis escolhidos para o projeto.
- b) Relacionar os elementos que facilitam ou dificultam o uso dos sensores aos conceitos científicos.
- c) Explorar o uso dos sensores adequados no desenvolvimento do objeto de aprendizagem.

1.4 Organização do trabalho

Esta monografia é dividida em quatro capítulos, sendo que no primeiro tem-se a introdução, as considerações a respeito do tema e problema estudado, objetivos gerais e específicos e a organização do trabalho.

No segundo capítulo são abordados os assuntos estudados para o desenvolvimento do projeto. Na primeira seção sobre os objetos de aprendizagem são apresentadas a definição, aspectos pedagógicos e exemplos de objetos. Logo em seguida tem-se a plataforma, arquitetura e motivos pelos quais o sistema operacional Android foi escolhido para o desenvolvimento do projeto. Na terceira seção é realizado um estudo detalhado de todos os sensores disponíveis na plataforma Android, bem como descrição dos dados obtidos, modo de utilização dos mesmos e tratamento de erros. Na última e quarta seção, são apresentados os conceitos físicos sobre a fotossíntese, tais como definição, histórico, importância, processos físicos e químicos relacionados.

O terceiro capítulo aborda os passos de desenvolvimento do projeto. Nele, são descritos os processos envolvidos no estudo e escolha do tema do objeto de aprendizagem, materiais e ferramentas utilizadas, interface do usuário, implementação, testes e resultados obtidos.

Por fim, no quarto e último capítulo, tem-se a conclusão e recomendações para projetos futuros.

2. REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os fundamentos teóricos utilizados para o desenvolvimento do projeto. Inicialmente, as definições para os objetos de aprendizagem e quais são os benefícios que estes materiais podem proporcionar para o ensino-aprendizagem de conceitos científicos.

Na seção seguinte será feita uma contextualização do sistema operacional Android, e apresentadas as vantagens e os motivos pelos quais este foi escolhido para desenvolvimento.

Logo em seguida, serão explorados os conceitos relacionados aos sensores e descritos quais destes encontra-se disponíveis para usuários do sistema Android e como utilizá-los.

2.1 Objetos de Aprendizagem

Com a criação da internet e os avanços tecnológicos da informática, os materiais didáticos utilizados no processo de ensino-aprendizagem passaram a contar com recursos multimídia que propiciaram maior interatividade, permitindo que o aprendizado se torne cada vez mais eficaz. (DIAS et al., 2009, p. 1).

“As Tecnologias da Informação e Comunicação (TICs) vêm sendo cada vez mais utilizadas na educação”. (ZANETTE; NICOLEIT; GIACOMAZZO, 2006, p. 1). Segundo Assis et al. (2011, p. 1158),

[...] o uso de TICs no ensino de Ciências contribui para o surgimento de novas práticas pedagógicas. A combinação da característica iconográfica, o uso de imagens e linguagem hipertextual se torna atrativo para a educação, especialmente quando se considera a transposição de fenômenos do meio natural para o meio digital. O uso de analogias permite que o aluno faça previsões e simultaneamente observe os efeitos das alterações das variáveis, contribuindo dessa forma para a construção de conceitos.

Inserido neste contexto, os objetos de aprendizagem (OA) vêm se mostrando como uma alternativa aos professores no apoio ao processo ensino-aprendizagem. (VIEIRA; NICOLEIT, 2007).

2.1.1 Definição

De acordo com o Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), um OA é “[...] qualquer entidade, digital ou não digital, que possa ser usada, na aprendizagem, educação e treinamento” (IEEE, 2002, tradução nossa).

A Rede Interativa Virtual de Educação (RIVED) define um objeto de aprendizagem como:

[...] qualquer recurso que possa ser utilizado para dar suporte ao aprendizado. Sua principal ideia é “quebrar” o conteúdo educacional disciplinar em pequenos trechos que podem ser reutilizados em vários ambientes de aprendizagem. Qualquer material eletrônico que provém informações para a construção de conhecimento pode ser considerado um objeto de aprendizagem, seja essa informação em forma de uma imagem, uma página HTML, uma animação ou simulação. (RIVED, [200-?])

Para Wiley (2001, p.3), um objeto de aprendizagem é “[...] qualquer recurso digital que possa ser utilizado para o suporte ao ensino”. Segundo Muzio, Heins e Mundell (2001) apud Silva (2006, p.101) "os OA são considerados objetos designados e/ou utilizados para fins instrucionais e incluem desde mapas e gráficos até demonstrações em vídeos e simulações interativas, na maioria das vezes de cunho digital". Pode ser definido também como:

Qualquer recurso, suplementar ao processo de aprendizagem, que pode ser reusado para apoiar a aprendizagem. O termo objeto educacional (*learning object*) geralmente aplica-se a materiais educacionais projetados e construídos em pequenos conjuntos com vista a maximizar as situações de aprendizagem onde o recurso pode ser utilizado. (...) A ideia básica é a de que os objetos sejam blocos com os quais será construído o contexto de aprendizagem (TAROUCO; FABRE; TAMUSIUNAS, 2003, p.2).

De acordo com Quinn apud Wilges (2006, p. 38), o modelo de objetos de aprendizagem pode ser caracterizado

[...] pela crença de que podemos criar pedaços independentes de conteúdos educacionais que proveja experiências educacionais para algum propósito educacional. Projetados sobre o modelo de programação orientada a objetos, este modelo assume que esses pedaços são autocontidos, que podem conter referências a outros objetos, e que podem ser combinados ou sequenciados para formar interações educacionais longas. Esses pedaços de conteúdo educacional pode ser de qualquer tipo – passivo, ativo – e pode ser de qualquer formato ou tipo de mídia. Um objeto de aprendizagem não é necessariamente um arquivo digital.

Apesar de não haver um consenso sobre a definição exata de um objeto de aprendizagem, vários autores concordam que um conteúdo pode ser considerado

como um OA caso o mesmo seja digital (vídeos, animações, simulações) e possua um conteúdo que facilite a aprendizagem em diferentes contextos. (DIAS et al., 2009).

[...] existem muitas diferentes definições para Objetos de Aprendizado e muitos outros termos são utilizados. Isto sempre resulta em confusão e dificuldade de comunicação, o que não surpreende devido a esse campo de estudo ser novo. (MUZIO apud BETTIO (2003, p. 17)).

Os OA devem ser projetados de maneira a integrarem-se com outros OA semelhantes, possibilitando a criação de contextos maiores. Ou seja, “são módulos que tem o objetivo de se ligarem e possibilitar a construção do conhecimento”. (VIEIRA; NICOLEIT, 2008, p. 142). Conforme sintetizado por Tarouco, Fabre e Tamusiunas (2003), um OA deve apresentar as seguintes características:

- Reusabilidade: um objeto de aprendizagem deve possuir um padrão para facilitar sua reutilização, ou seja, deve possuir requisitos que possibilitem usuários diferentes, em contextos diferentes a utilizá-lo.
- Interoperabilidade: De acordo com Arantes, Mirando e Studart (2010, p. 28), “refere-se a capacidade de o sistema operar através de uma variedade de hardware, sistemas operacionais e buscadores”. Assim, é possível utilizar o mesmo objeto em locais distintos utilizando diferentes ferramentas e plataformas.
- Acessibilidade: facilmente acessível via internet;
- Durabilidade: capacidade de se adaptar caso a tecnologia de base seja alterada, sem a necessidade de reprogramação do objeto de aprendizagem, já que o mesmo pode se adaptar a qualquer plataforma;
- Adaptabilidade: garante que o objeto de aprendizagem será adaptado para tratar situações e necessidades individuais, ou seja, é adaptável a qualquer ambiente de ensino;
- Granularidade: quanto maior a granularidade, maior será sua reutilização. A granularidade define o tamanho ideal do objeto de aprendizagem;

2.1.2 Aspectos Pedagógicos

Segundo o RIVED ([200-?]), os OAs visam estimular o raciocínio e pensamento crítico dos estudantes.

[...] a utilização de OAs possibilita ao aluno conhecer ferramentas novas, testar diferentes situações, arriscar, antecipar, compreender a relação entre causa e efeito de conceitos, fenômenos, para despertar a curiosidade e para resolver problemas. (VIEIRA; NICOLEIT, 2008, p.142).

Para alcançar os objetivos propostos pelos OAs, é necessário que o processo de aprendizagem agregue três fatores importantes (VIEIRA; NICOLEIT, 2008):

- O conteúdo esteja estruturado de maneira compreensível;
- O aluno tenha conhecimento prévio que lhe possibilite entender o assunto a ser abordado;
- Aceitação do aluno por esse novo tipo de ensino, relacionando-o com seus conhecimentos prévios.

Objetivos esses que só serão atingidos caso o objeto de aprendizagem possibilite ao aluno agregar o seu conhecimento básico com o conhecimento avançado proposto. Para isso, pode-se utilizar recursos visuais como animações e simulações que atraiam a atenção do aluno e criem um ambiente onde ele possa vivenciar o tema abordado. (BORGES; NAVARRO, 2005).

2.1.3 Modelo ADDIE para o desenvolvimento de Objetos de Aprendizagem

O desenvolvimento de um software bem estruturado depende de uma metodologia de trabalho definida, capaz de oferecer a base para a construção do mesmo de maneira eficaz. Com o objetivo de garantir a criação de um OA organizado, com níveis de variações mínimas e dentro de níveis aceitáveis é necessário aplicar técnicas de engenharia de software. (LUCENA et al., 2010).

Mustaro et al. (2007) apud Pessoa e Benitti (s.d, p.2) “sugere a utilização do modelo ADDIE (acrônimo para Analysis, Design, Development, Implementation e

Evaluation) para o desenvolvimento de Objetos de Aprendizagem”. Segundo Pessoa e Benitti (s.d) este modelo compreende as seguintes etapas, conforme mostra a Figura 1:



Figura 1 - Etapas do desenvolvimento do modelo ADDIE

FONTE: Elaborado pela autora com base na imagem disponível no site Teaching Online¹

- Análise (*Analysis*): caracterizada pela determinação dos objetivos a serem alcançados (principalmente educacionais) do OA, bem como seus pré-requisitos, reusabilidade, cenário tecnológico e mídias apropriadas;

- Projeto (*Design*): todas as informações coletadas na etapa anterior são utilizadas para o planejamento da etapa de Projeto. Nela, ocorre a definição dos pontos importantes sobre o design de interfaces como usuário final, sequência dos conteúdos, interatividade e *feedback*.

- Desenvolvimento (*Development*): consiste no desenvolvimento, propriamente dito, do objeto de aprendizagem. Nela, podem ocorrer atividades como a produção de elementos de mídia, testes de usabilidade, interface e navegação.

- Implementação (*Implementation*): compreende a execução do OA, ou seja, a publicação e execução de tudo o que já foi desenvolvido. No caso de um material desenvolvido para o ensino à distância, é nessa fase que os participantes utilizarão as atividades criadas.

¹ Disponível em <<http://teach.ucf.edu/files/2010/04/2.jpg>>. Acesso em: 10 dez. 2013.

- *Avaliação (Evaluation)*: nesta última etapa é feita uma avaliação das funcionalidades do OA gerado, ou seja, se estas condizem com a especificação de requisitos inicialmente levantadas, a eficácia e eficiência dos métodos, técnicas e ferramentas empregadas. (GOTARDO, 2012).

Apesar de contemplar mais fases e conter algumas diferenças em suas etapas, é importante ressaltar que o Modelo Cascata, descrito por Pressman (2011) se assemelha muito ao Modelo ADDIE, por contemplar etapas de análise, projeto, desenvolvimento e avaliação (Figura 2).

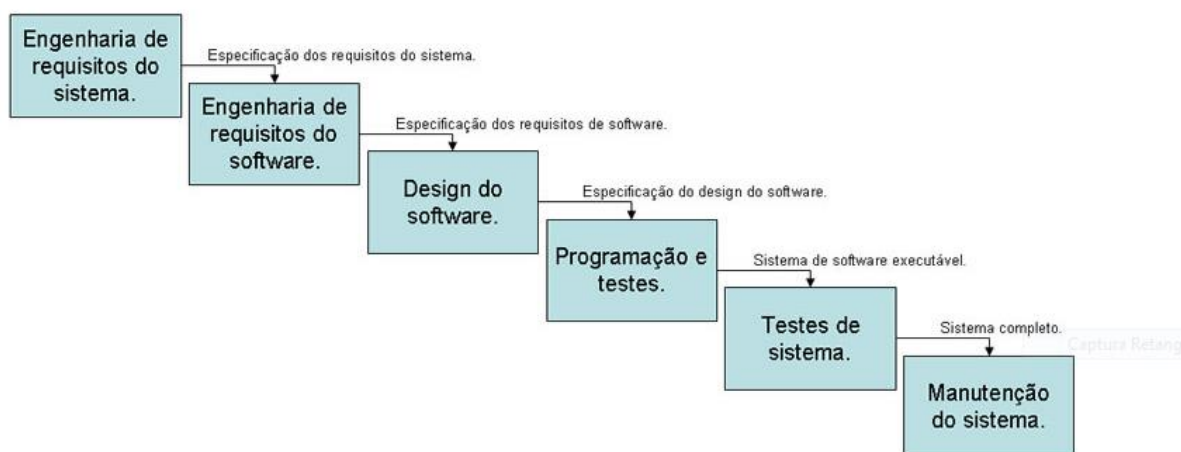


Figura 2 - Ciclo de Vida do Processo de Desenvolvimento de Software (usando o Modelo Cascata)
 FONTE: Disponível no site da Faculdade do Porto – FEUP (Faculdade de Engenharia).²

2.1.4 Objetos de Aprendizagem no Brasil

De acordo com Castro (s.d, p. 2), “a criação de objetos de aprendizagem voltada para o ensino tem crescido bastante”. Nos próximos tópicos serão apresentados os repositórios de OA disponíveis no Brasil e projetos que aliam a utilização dos mesmos aos dispositivos móveis.

2.1.4.1 Repositórios de Objetos de Aprendizagem

² Disponível em <<http://twiki.fe.up.pt/bin/view/ERSS0506/ModelosERSoftware>>. Acesso em: 15 dez. 2013.

No Brasil, existem dois principais repositórios de OAs: o RIVED (Rede Interativa Virtual de Educação) e outro criado pela USP, o LabVirt (Laboratório Didático Virtual).

2.1.4.1.1 RIVED

O RIVED disponibiliza acesso a vários OAs e foi fundado em 1999, por meio da parceria entre Secretaria de Educação a Distância (SEED), do Ministério da Educação (MEC), e Secretaria de Educação Média e Tecnológica (SEMTEC). Em 2004 a produção de objetos de aprendizagem passou a ser realizada por universidades e essa ação recebeu o nome de Fábrica Virtual.

O projeto tem por objetivo a produção de conteúdos pedagógicos digitais para o Ensino Médio, Ensino Fundamental, Ensino Profissionalizante, Superior. Os OAs produzidos pelo RIVED são atividades multimídia (animações e simulações), interativas.

A possibilidade de testar diferentes caminhos, de acompanhar a evolução temporal das relações, causa e efeito, de visualizar conceitos de diferentes pontos de vista, de comprovar hipóteses, fazem das animações e simulações instrumentos poderosos para despertar novas ideias, para relacionar conceitos, para despertar a curiosidade e para resolver problemas. Essas atividades interativas oferecem oportunidades de exploração de fenômenos científicos e conceitos muitas vezes inviáveis ou inexistentes nas escolas por questões econômicas e de segurança, como por exemplo: experiências em laboratório com substâncias químicas ou envolvendo conceitos de genética, velocidade, grandeza, medidas, força, dentre outras. (RIVED, [200-?])

Um desses OA denomina-se *Trigonometria na Ponte* (Figura 3) e tem por objetivo apresentar, de forma intuitiva o conceito da Lei dos Senos. O objeto é composto por três atividades.

- Atividade 1: é apresentada uma situação onde o aluno deve calcular a distância de um lado a outro de uma lagoa, para que uma ponte que ligue uma estrada possa ser construída. O objetivo dessa atividade é fazer com que o aluno aplique o Teorema de Pitágoras para calcular o comprimento da ponte.
- Atividade 2: o problema proposto por essa atividade não pode mais ser resolvido com a aplicação do Teorema de Pitágoras. Para encontrar a resolução para este caso o aluno deverá utilizar Razões Trigonométricas.

- Atividade 3: nesta última atividade o problema só poderá ser resolvido com a aplicação da Lei dos Senos.

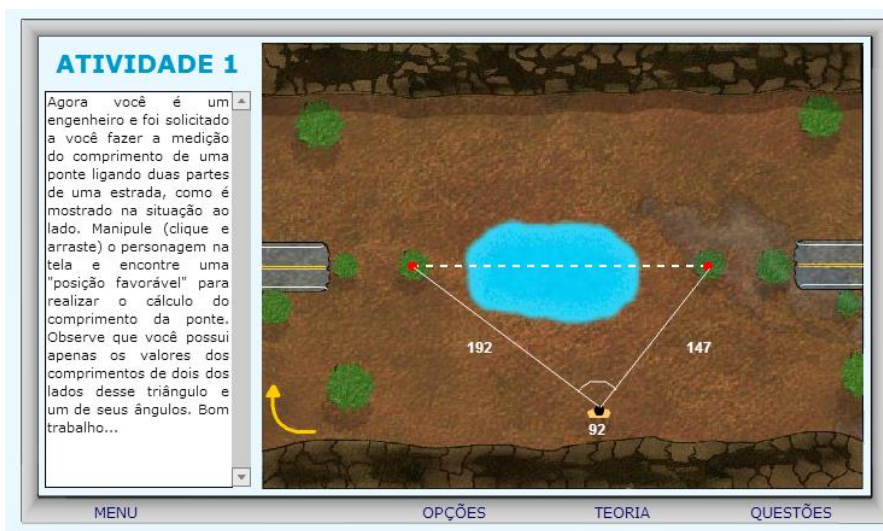


Figura 3 - Figura ilustrativa do objeto Trigonometria na ponte
 FONTE: Disponível no site RIVED.³

No objeto de aprendizagem *Futebol no país da matemática* (Figura 4) o foco é mostrar ao aluno a relação existente entre o comprimento de um arco de circunferência e o ângulo a partir da cobrança de pênaltis de futebol. É composto por três atividades:

- Atividade 1: apresenta conceitos básicos de como cobrar um pênalti e estabelece quais são as similaridades existentes entre o futebol e a matemática.
- Atividade 2: nesta atividade o aluno trabalha com a distância do jogador em relação ao gol.
- Atividade 3: dois campos de futebol, com medidas distintas, são apresentados e o aluno deve aplicar o conhecimento adquirido nas atividades anteriores para realizar a cobrança de pênalti.

³ Disponível em http://rived.mec.gov.br/atividades/matematica/trigonometria_ponte/mat1_ativs1.html >. Acesso em: 31 maio 2013.

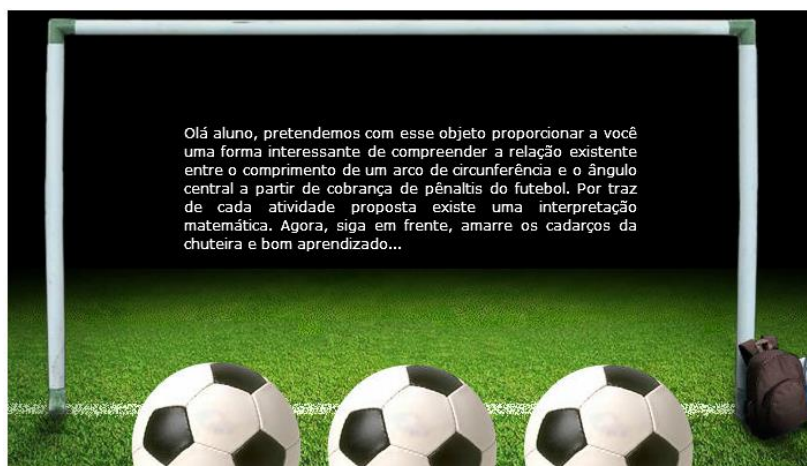


Figura 4 - Figura ilustrativa do objeto Futebol no país da matemática⁴
 FONTE: Disponível no site RIVED.

Em *Aprendendo as Leis de Newton com os carrinhos de rolimã* (Figura 5) o objetivo é auxiliar o processo de aprendizagem dos conceitos de física relacionados às Leis de Newton, tais como massa, peso, força e atrito.

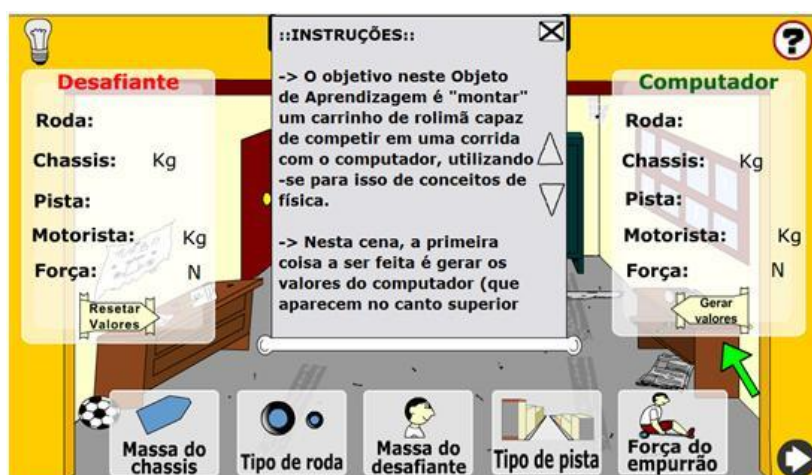


Figura 5 - Figura ilustrativa do objeto Aprendendo as Leis de Newton.
 FONTE: Disponível no site Banco Internacional de Objetos de Aprendizagem.⁵

2.1.4.1.2 LabVirt

O *LabVirt* é um repositório de OAs criado pela Universidade de São Paulo (USP). Contém vários objetos do tipo animação e simulação nas áreas de física e química.

⁴ Disponível em < http://rived.mec.gov.br/atividades/matematica/o_futebol/mat1_intro.html>. Acesso em: 31 maio 2013.

⁵ Disponível em < <http://objetoseducacionais2.mec.gov.br/handle/mec/966> >. Acesso em: 31 maio 2013.

A simulação *A química dentro de um bolo* (Figura 6) aplica conceitos de balanceamento de reações químicas, por meio do processo de preparação de um bolo.

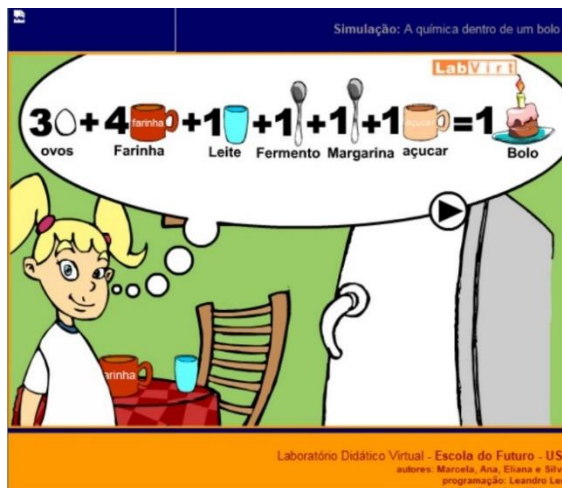


Figura 6 - Figura ilustrativa do objeto *A química dentro de um bolo*
 FONTE: Disponível no site *A graça da Química*.⁶

Ondas de rádio (Figura 7) é uma outra simulação disponibilizada pelo *LabVirt*. Esta atividade apresenta a história de uma família que estava viajando e ouvindo música e em determinado momento, o rádio sai do ar e começa a emitir chiados e interferência de outras rádios. O objetivo deste objeto é que o aluno seja capaz de calcular o comprimento de onda de uma faixa de frequência de rádio e ajudar a família em questão a continuar ouvindo música durante o trajeto.



Figura 7 - Figura ilustrativa do objeto *Ondas de Rádio*
 FONTE: Disponível no site *LabVirt*.⁷

⁶ Disponível em < <http://www.agracadaquimica.com.br/simulacoes/28.swf> >. Acesso em: 31 maio 2013.

⁷ Disponível em < http://www.labvirt.fe.usp.br/simulacoes/fisica/sim_eletro_ondasderadio.htm >. Acesso em: 31 maio 2013.

2.1.4.2 Objetos de Aprendizagem para dispositivos móveis

O desenvolvimento de Objetos de Aprendizagem para dispositivos móveis pode ser encontrado em várias pesquisas. Dentre elas, pode-se citar: Cardoso et al. (2013) - desenvolvimento de um OA Móvel que aborda o ensino de Vetores, denominado OAMVet; Filho e Waechter (2013) - desenvolvimento de um OA, em forma de infográfico interativo para *tablets*, sobre o peixe-boi; Oliveira (2013) - OA sobre o tema da divisão celular, mais especificamente mitose. (Figura 8).



Figura 8 - Exemplos de Objetos de Aprendizagem para dispositivos móveis
FONTE: Elaborada pela autora com base nas obras de Cardoso et al. (2008), Filho e Waechter (2013) e Oliveira (2013).

Além das pesquisas citadas, é possível citar a EvoBooks, uma editora digital de conteúdo educacional para *tablets* (Android e iOS). A coleção desenvolvida pela empresa traz aulas interativas adaptadas ao Ensino Básico de Ciências (Biologia, Química e Física), História e Geografia.

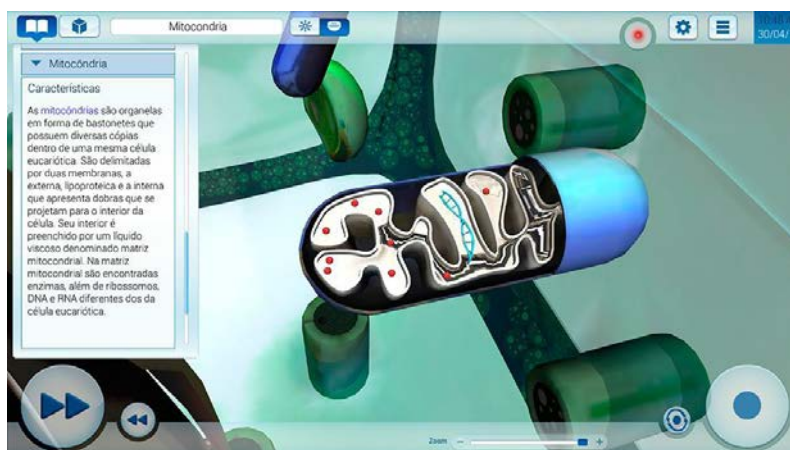


Figura 9 - Exemplo de um objeto de aprendizagem sobre Biologia, desenvolvido pela EvoBooks
FONTE: Disponível na página do aplicativo Biologia Celular EvoBooks, no Google Play.⁸

⁸ Disponível em
<<https://lh3.ggpht.com/DCSgiUm1Xwo2RANWWOpYmwjv6jcl5w2ZraHqgCbDcgNh4lqc2T8nKpsTd874X1XUNs0=h900>>. Acesso em: 31 maio 2013.

2.2 Sistema Operacional Android

2.2.1 Plataforma

O Android foi desenvolvido pela empresa Android Inc., empresa fundada em 2003 por Andy Rubin, Rich Miner, Nick Sears e Chris White, na cidade de Palo Alto na Califórnia (EUA). Trata-se de um sistema operacional de código aberto (*open source*) projetado principalmente para *smartphones* e *tablets*. Contém um sistema operacional baseado em Linux, interface visual rica, diversas aplicações disponíveis e um ambiente de desenvolvimento bastante poderoso e flexível.

A trajetória do sistema tem início em agosto de 2005, quando a Android Inc. foi comprada pela Google. Logo em seguida, em novembro de 2007, a Google estabeleceu uma aliança comercial (Open Handset Alliance – OHA) com empresas líderes do mercado de telefonia visando padronizar uma plataforma de código aberto e livre para celulares, justamente para atender a todas as expectativas e tendências do mercado. Entre alguns integrantes dessa aliança estão a Motorola, LG, Samsung, Sony Ericsson, Toshiba, Sprint Nextel, HTC, China Mobile, T-Mobile, ASUS e Intel.

De acordo com Lecheta (2010), o objetivo do grupo OHA é definir uma plataforma única e aberta para celulares para oferecer um produto final com qualidade, criar uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas.

A primeira geração de telefones Android foi lançada em outubro de 2008. A partir daí, o sistema recebeu uma série de atualizações do Google, cada uma oferecendo novos recursos, funcionalidades e melhoria de desempenho. A última versão do sistema foi lançada em outubro de 2013, Android 4.4 (KitKat).

Conforme dados atualizados pela Google em dezembro de 2013, a Figura 10 representa a atual adoção das versões Android presente nos dispositivos móveis. Os dados apresentados não consideraram versões inferiores ao Android 2.2, pois elas não oferecem suporte ao novo aplicativo *Play Store*, que é responsável por avaliar a quantidade de aparelhos que visitam a loja.

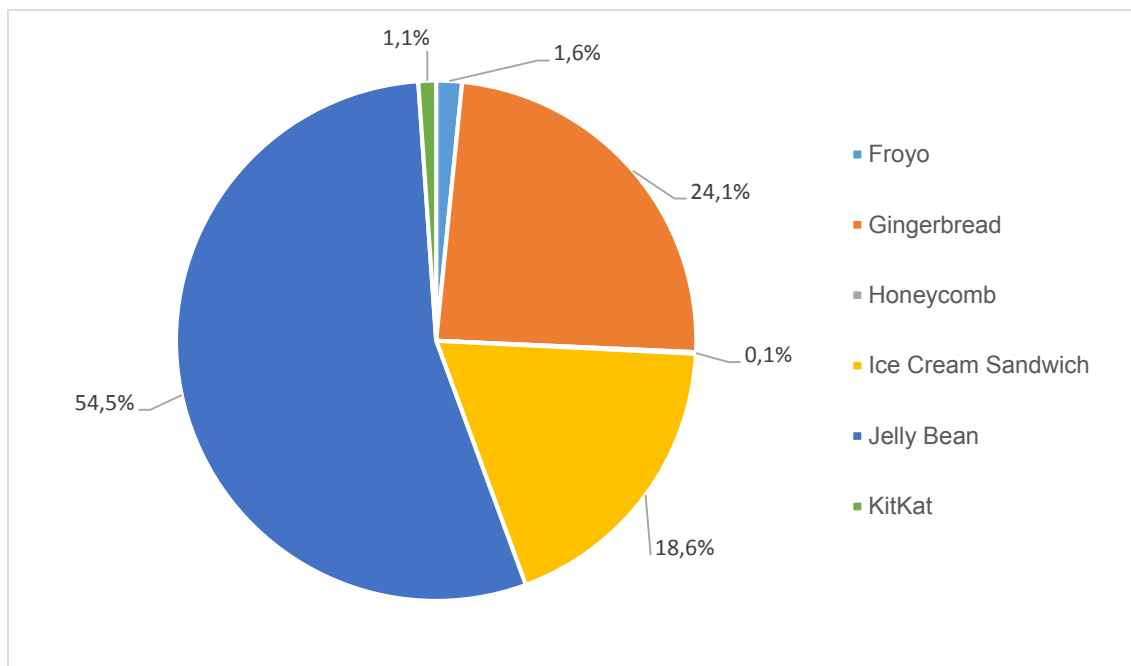


Figura 10 - Atual adoção das versões Android presentes nos dispositivos móveis
FONTE: (ALMEIDA, 2013).

2.2.2 Arquitetura do sistema operacional Android

O Google geralmente se refere ao sistema operacional Android como uma pilha de softwares. Cada camada desta pilha é composta por vários programas que desempenham funções específicas do sistema operacional.

Pereira e Silva (2009) apresentam a arquitetura Android estruturada em cinco camadas (Figura 11).

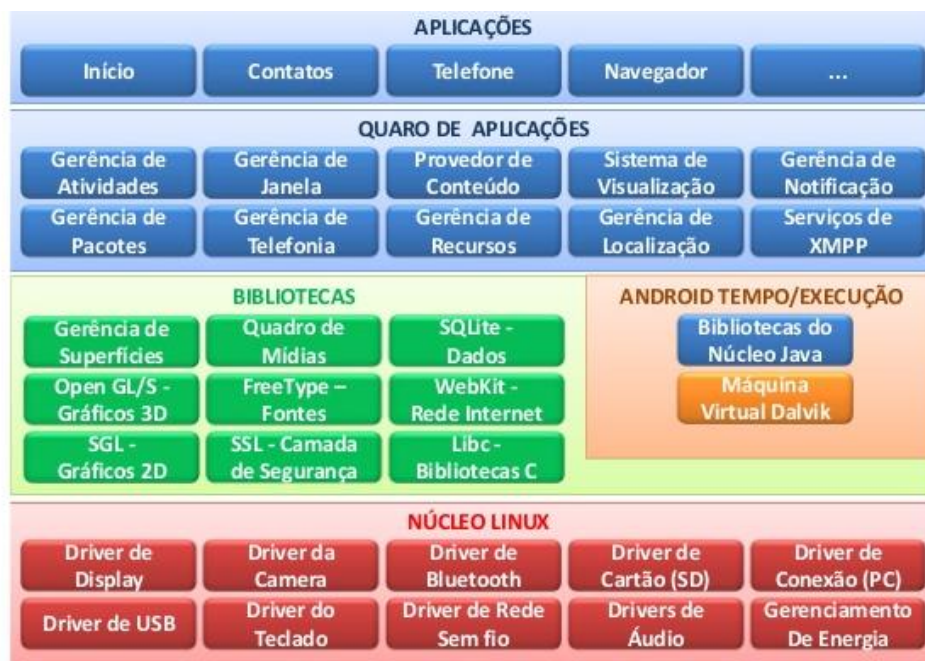


Figura 11 - Arquitetura da plataforma Android
 FONTE: (OLIVEIRA, 2013).

2.2.2.1 Kernel – Núcleo Linux

A base da pilha é o kernel do sistema. Nesta camada é utilizada a versão 2.6 do kernel do Linux para os serviços centrais do sistema, que incluem gerenciamento de energia, segurança, gestão de memória, de processos e de vários drivers de hardware. Além disso o kernel atua como uma camada de abstração entre o hardware e as outras camadas da pilha de software.

2.2.2.2 Bibliotecas

A camada de bibliotecas é constituída por um conjunto de bibliotecas em C/C++, que podem ser acessadas através dos frameworks disponibilizados para desenvolvedores. Incluem bibliotecas multimídia, visualização das camadas 2D e 3D, funções para navegadores web, gráficos, aceleração de hardware, renderização 3D, fontes bitmap e vetorizadas e funções de acesso ao banco SQLite.

2.2.2.3 Android Runtime (Ambiente de Execução)

A camada do ambiente de execução corresponde a uma instância da máquina virtual Dalvik, criada para cada aplicação executada no sistema. Isso é importante pelas seguintes razões: as aplicações se tornam independentes umas das outras; um possível erro de execução em alguma das aplicações rodando no dispositivo não afeta o sistema como um todo e por fim, a simplificação do gerenciamento de memória.

2.2.2.4 Framework

A camada do framework engloba todas as APIs e recursos utilizados pelos aplicativos, atuando como o intermediário entre a camada de aplicações e a camada de bibliotecas.

2.2.2.5 Aplicações

A camada no topo da pilha corresponde aos aplicativos em si (escritos em linguagem Java), na qual encontra-se um conjunto de aplicações, tais como navegador de internet, mapas, despertador e acesso a lista de contatos, por exemplo.

2.2.3 Desenvolvimento de software para Android

2.2.3.1 Mercado Mundial

De acordo com pesquisas realizadas pela consultoria IDC, no terceiro semestre de 2013 as vendas de *smartphones* atingiram cerca de 468 milhões de unidades vendidas, um aumento de 39% em relação ao mesmo período do ano passado (Figura 12).

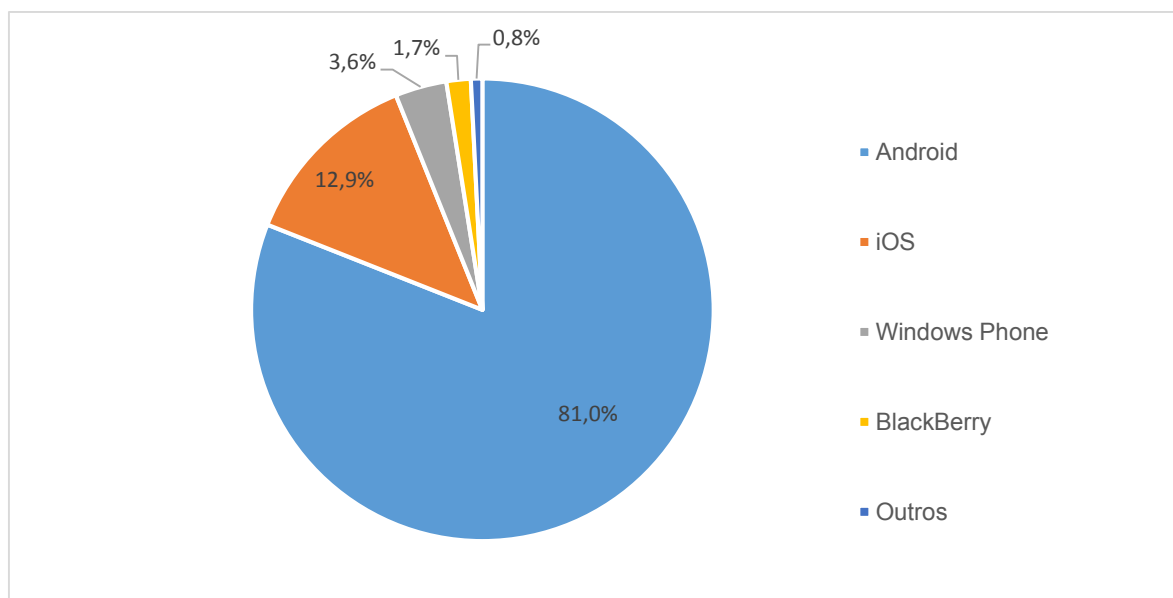


Figura 12 - Segmento dos sistemas operacionais para *smartphones* - participação de mercado no final do terceiro trimestre de 2013.
 FONTE: (IDC, 2013).

Segundo dados da Strategy Analytics Tablets Service (2013), entre os três segmentos de dispositivos móveis, os *tablets* foram os que registraram o maior crescimento neste ano. A venda global de *tablets* teve um aumento de 47% (36,2 milhões aparelho vendidos) em relação ao segundo trimestre de 2012, onde foram comercializados 24,6 milhões de aparelhos (Figura 13).

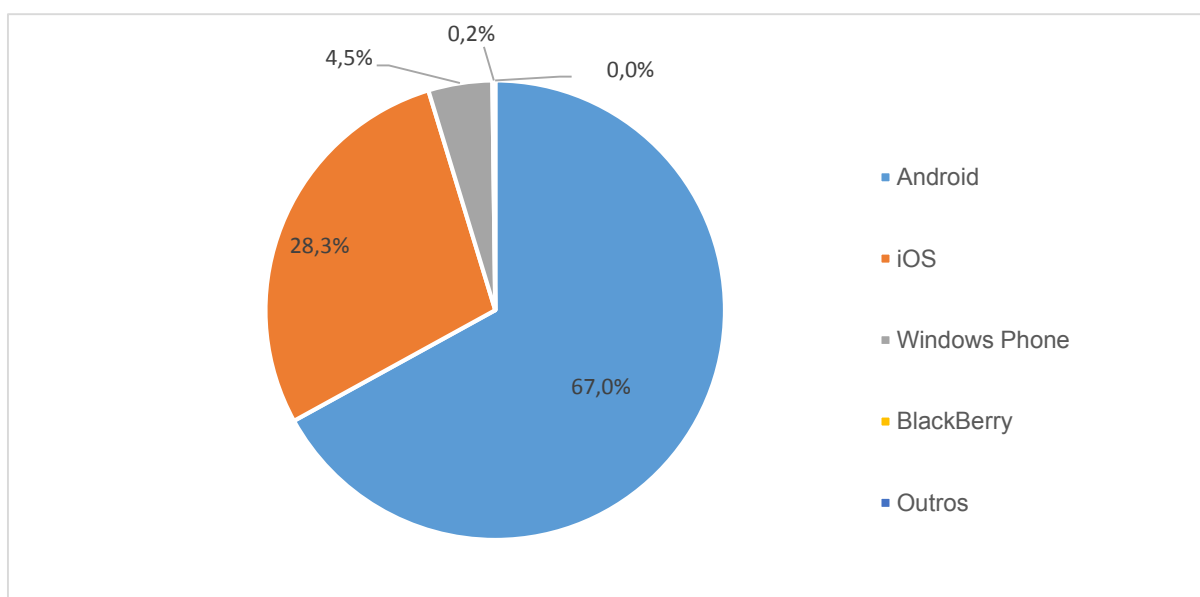


Figura 13 - Segmento dos sistemas operacionais para *tablets* - participação de mercado no final do terceiro trimestre de 2013.
 FONTE: (STRATEGY ANALYTICS TABLETS SERVICE, 2013).

2.2.3.2 Código aberto e livre

O Android é completamente livre e de código aberto (open source), o que proporciona uma evolução constante ao sistema, já que diversos programadores poderão contribuir para torná-lo melhor. Isso também favorece os fabricantes de celulares, uma vez que é possível criar novas interfaces para o usuário e adicionar recursos internos, sem a necessidade de ter que pagar para usar o sistema. (LECHETA, 2010).

A documentação disponibilizada pelo Android tem facilitado a navegação em vários pacotes presentes no SDK. Inclui também um Guia de Desenvolvimento de alto nível e links para a comunidade Android.(IBM, s.d). Pelo fato de ser open source há vários tutoriais disponíveis em vídeos, sites e comunidades que facilitam o aprendizado e desenvolvimento de aplicativos Android.

2.2.3.3 Android SDK (Kit de Desenvolvimento de Software)

O Android SDK é o software utilizado para desenvolver aplicações no Android, com suporte às plataformas Windows, Mac OS X e Linux. Contém um emulador para simular o dispositivo, ferramentas utilitárias e uma API completa para a linguagem Java, com todas as classes necessárias para o desenvolvimento das aplicações. (LECHETA, 2010).

Essas aplicações são escritas em linguagem Java utilizando o IDE (*Integrated Development Environment*) do Eclipse, que disponibiliza um *plug-in* chamado ADT (*Android Developer Tools*) para facilitar o desenvolvimento, testes e compilação do projeto.

Embora a maioria das aplicações seja escritas em Java, algumas pequenas partes do *framework* não são. O sistema também inclui a linguagem XML (*Extensible Markup Language*). Portanto, os requisitos básicos para começar a desenvolver uma aplicação Android envolvem o conhecimento da linguagem Java e XML.

2.2.3.4 Dispositivos móveis

O Android pode ser executado em vários dispositivos com diferentes tamanhos de tela e resoluções. Ao contrário do iOS, por exemplo, que possui seu próprio sistema operacional exclusivo para o hardware do dispositivo, o Android contém ferramentas que auxiliam o desenvolvimento de aplicações compatíveis com diferentes fabricantes. Entre eles estão a Motorola, Sony, Samsung e LG, cada um com uma adaptação para o sistema e com diferentes versões Android.

2.2.3.5 Recursos de Hardware

A Google disponibiliza uma infinidade de funcionalidades do Android, oferecendo aos desenvolvedores recursos de hardware necessários para a criação de aplicativos completos (Quadro 1).

Quadro 1 - Recursos de Hardware do Android

Funcionalidade necessária	Hardware
Onde estou?	GPS
Em qual direção estou caminhando?	Bússola
O meu aparelho está em movimento?	Acelerômetro
Posso usar meu fone de ouvido Bluetooth?	Bluetooth
Como eu posso gravar um vídeo?	Câmera

FONTE:(FELKER; DOBS, 2011, p.18).

Com a evolução dos dispositivos móveis, alguns fabricantes de hardware podem adicionar recursos que ainda não são suportados nativamente pelo Android. Geralmente quando isso ocorre, estes fabricantes disponibilizam um SDK para que os desenvolvedores usufruíssem das características únicas de tais dispositivos.

2.3 Sensores internos disponíveis nos dispositivos Android

2.3.1 Sensores

Segundo Fuentes (2005), um sensor pode ser definido como um dispositivo que converte uma grandeza física (como temperatura, pressão, umidade, velocidade, aceleração e luminosidade) em um sinal elétrico (Figura 14).

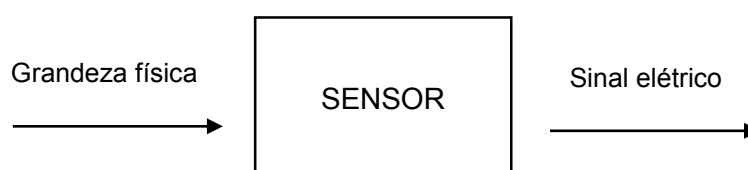


Figura 14 - Esquema de funcionamento de um sensor

FONTE: Adaptação da imagem disponível em FUENTES, s.d, p. 1.

As principais características de um sensor são: *linearidade* (representa o grau de proporcionalidade entre o sinal gerado e a grandeza física). Quanto maior a linearidade, mais fiel é a resposta do sensor ao estímulo recebido); e *faixa de atuação* (representa o intervalo de valores da grandeza em que pode ser utilizado o sensor, sem causar sua destruição ou a imprecisão na leitura). (ROSÁRIO, 2005).

De acordo com Nomads ([200-?]), os sensores podem ser classificados de acordo com o tipo de energia que detectam. Por exemplo:

- *Sensores de luz*: células solares, fotodiodos, fototransistores, tubos fotoelétricos, sensor de imagem;
- *Sensores de som*: microfone, sensores sísmicos;
- *Sensores de temperatura*: termômetros, termopares, resistências sensíveis a temperatura, termostatos;
- *Sensores de calor*: bolómetro, calorímetro;
- *Sensores de radiação*: contador Geiger, dosímetro;
- *Sensores de resistência elétrica*: ohmímetro;
- *Sensores de corrente elétrica*: galvanômetro, amperímetro;
- *Sensores de tensão elétrica*: electrómetro, voltímetro;
- *Sensores de potência elétrica*: wattímetro;

- *Sensores magnéticos*: bússola magnéticas, bússola de fluxo de porta, magnetômetro, dispositivo de efeito Hall;
- *Sensores de pressão*: barômetro, barógrafo, indicadores da velocidade do ar;
- *Sensores de movimento*: velocímetro, coordenador de giro, tacômetro;
- *Sensores de orientação*: giroscópio, horizonte artificial, giroscópio de anel de laser;
- *Sensores mecânicos*: sensor de posição, selsyn, chave;
- *Sensores de proximidade*: presentes em telefones celulares, detecção de papel nas fotocopiadoras, por exemplo.

Antes do lançamento dos dispositivos móveis os sensores eram raramente encontrados no cotidiano das pessoas. Geralmente estavam presentes em aparelhos projetados para propósitos específicos, tais como medir a temperatura em fornos, pressão de pneus e sistema de controle remoto de televisões.

A partir do Android 1.5, *tablets* e *smartphones* começaram a fornecer suporte a uma série de sensores, oferecendo a usuários e desenvolvedores a possibilidade de utilizá-los para finalidades diversas, incluindo realidade aumentada e entrada baseada em movimentos.

2.3.2 Tipos de sensores internos disponíveis nos dispositivos Android

Os sensores presentes nos dispositivos Android são baseados em dois tipos (Android Developers, 2013d) (Quadro 2):

- *Hardware*: são componentes físicos presentes em *smartphones* e *tablets*. Oferecem dados obtidos através de medições realizadas diretamente sobre determinadas propriedades ambientais, tais como aceleração, intensidade do campo magnético ou mudança angular.
- *Software ou Sintéticos*: não são componentes físicos. Na verdade, fornecem uma camada de abstração entre o código da aplicação e os componentes de baixo

nível dos aparelhos móveis. Os dados obtidos provêm da combinação de um ou mais sensores baseados em hardware.

Quadro 2 - Tipos de sensores suportados pela plataforma Android

Sensor	Tipo	Descrição	Uso
TYPE_ACCELEROMETER	Hardware	Retorna a força de aceleração (em m/s^2) aplicada ao longo dos três eixos (x, y e z) do aparelho, incluindo a força da gravidade.	Detecção de movimento (balançar, inclinar)
TYPE_AMBIENT_TEMPERATURE	Hardware	Retorna a temperatura ambiente em graus Celsius ($^{\circ}C$)	Monitorar a temperatura do ar
TYPE_GRAVITY	Software ou Hardware	Retorna a força da gravidade (em m/s^2) aplicada ao longo dos três eixos (x, y e z).	Detecção de movimento (balançar, inclinar)
TYPE_GYROSCOPE	Hardware	Retorna a taxa de rotação aplicada ao aparelho em torno de cada um dos três eixos (x, y e z)	Detecção de rotação (por exemplo, girar)
TYPE_LIGHT	Hardware	Retorna o nível de luz ambiente (iluminação) em lx.	Controlar o brilho da tela
TYPE_LINEAR_ACCELERATION	Software ou hardware	Retorna a força de aceleração (em m/s^2) aplicadas ao longo dos três eixos (x, y e z) do aparelho, excluindo a força da gravidade.	Monitoramento da aceleração ao longo de um único eixo.
TYPE_MAGNETIC_FIELD	Hardware	Retorna o campo geomagnético ambiente para todos os três eixos (x, y e z) em μT	Criação de uma bússola.

TYPE_ORIENTATION	Software	Retorna os graus de rotação que o aparelho faz em torno dos três eixos (x, y e z). A partir da API nível 3 é possível obter a matriz de inclinação e rotação de um dispositivo utilizando o sensor de gravidade e o sensor de campo magnético, em conjunto com o método <i>getRotationMatriz()</i> .	Determinar a posição do dispositivo.
TYPE_PRESSURE	Hardware	Retorna a pressão do ar ambiente em hPa ou mbar.	Monitorar as mudanças de pressão do ar.
TYPE_PROXIMITY	Hardware	Retorna a proximidade (em cm) de um objeto em relação a tela do aparelho	Posicionamento do telefone durante o recebimento de uma chamada.
TYPE_RELATIVE_HUMIDITY	Hardware	Retorna a umidade relativa do ambiente em porcentagem (%)	Monitoramento da umidade relativa e absoluta do ambiente.
TYPE_ROTATION_VECTOR	Software ou hardware	Retorna a orientação do aparelho tal como uma combinação de um ângulo em torno de um eixo.	Deteção de movimento e detecção de rotação
TYPE_TEMPERATURE	Hardware	Retorna a temperatura interna do aparelho em graus Celsius (°C). A implementação deste sensor varia entre dispositivos e foi substituído com o sensor TYPE_AMBIENT_	Monitoramento de temperatura

		TEMPERATURE na API nível 14.	
--	--	---------------------------------	--

FONTE: Adaptação do quadro (*Table 1*) presente em Android Developers, 2013d.

2.3.3 Categorias

A plataforma Android suporta três categorias de sensores:

2.3.3.1 Sensores Ambientais

Composto por quatro sensores baseados em hardware que permitem monitorar várias propriedades ambientais. São utilizados para monitorar a umidade relativa do ambiente, iluminação, pressão atmosférica e temperatura ambiente. Com exceção do sensor de luz, normalmente empregado para controle do brilho da tela, nem sempre estão disponíveis nos dispositivos portáteis.

2.3.3.2 Sensores de movimento

Tais sensores permitem controlar o movimento realizado em um dispositivo. Dois deles são baseados em hardware (acelerômetro e giroscópio), e três podem ser baseados em hardware ou software (gravidade, aceleração linear e sensores de vetor de rotação). A maioria dos dispositivos Android possuem um acelerômetro, e muitos agora incluem um giroscópio. A disponibilidade dos sensores baseados em software é variável, pois muitas vezes dependem de um ou mais sensores de hardware para obter seus dados.

São úteis para indicar a movimentação realizada por um dispositivo, tal como inclinação, rotação ou agitação. O movimento é geralmente um reflexo de uma entrada direta do usuário (por exemplo, um jogador controlando a direção de um carro em um jogo), ou um reflexo do ambiente físico no qual o dispositivo está inserido (por exemplo, a movimentação que o mesmo está sofrendo enquanto o usuário está dirigindo o carro).

2.3.3.3 Sensores de orientação

A plataforma Android fornece dois sensores que permitem determinar a orientação de um dispositivo: o sensor de campo magnético e o sensor de orientação. Além desses, fornece um sensor que permite determinar o quão perto o aparelho está do rosto do usuário (utilizado por exemplo, durante uma ligação telefônica), normalmente presente em aparelhos celulares: sensor de proximidade.

Tanto o sensor de campo magnético quanto o sensor de proximidade são baseados em hardware. O sensor de orientação é baseado em software e deriva seus dados do acelerômetro e do sensor de campo magnético.

2.3.4 Disponibilidade

O número de sensores presentes nos dispositivos móveis pode variar entre versões Android. Isso se deve ao fato de que estes foram sendo introduzidos gradativamente à plataforma. O Android 1.5, por exemplo, adicionou vários deles ao sistema. Mas, alguns não haviam sido implementados e se encontravam indisponíveis para uso até o Android 2.3.

O Quadro 3 mostra quais são os sensores disponíveis para cada versão Android. São listadas apenas aquelas que trouxeram mudanças e introduziram novos sensores.

Quadro 3 - Disponibilidade de sensores de acordo com a versão do Android

Sensor	Android 4.0 (API nível 14)	Android 2.3 (API nível 9)	Android 2.2 (API nível 8)	Android 1.5 (API nível 3)
TYPE_ACCELEROMETER	Sim	Sim	Sim	Sim
TYPE_AMBIENT_TEMPERATURE	Sim	Não	Não	Não
TYPE_GRAVITY	Sim	Sim	Não	Não
TYPE_GYROSCOPE	Sim	Sim	Não(*)	Não(*)
TYPE_LIGHT	Sim	Sim	Sim	Sim
TYPE_LINEAR_ACCELERATION	Sim	Sim	Não	Não
TYPE_MAGNETIC_FIELD	Sim	Sim	Sim	Sim
TYPE_ORIENTATION	Sim(**)	Sim(**)	Sim (**)	Sim

TYPE_PRESSURE	Sim	Sim	Não (*)	Não(*)
TYPE_PROXIMITY	Sim	Sim	Sim	Sim
TYPE_RELATIVE_HUMIDITY	Sim	Não	Não	Não
TYPE_ROTATION_VECTOR	Sim	Sim	Não	Não
TYPE_TEMPERATURE	Sim (**)	Sim	Sim	Sim

FONTE: Adaptação do quadro (*Table 2*) presente em Android Developers, 2013d.

(*) Este tipo de sensor foi adicionado no Android 1.5 (API nível 3), mas não estava disponível para uso até o Android 2.3 (API nível 9)

(**) Este sensor está disponível, mas tem se tornado obsoleto.

2.3.5 Como utilizar os sensores

Através da API do Android é possível:

- Determinar quais sensores estão disponíveis em um aparelho;
- Determinar características de um sensor específico, tais como nome, fabricante, taxa de operação e tipo, por exemplo;
- Adquirir valores obtidos pelos sensores e determinar a taxa mínima de operação dos mesmos;
- Registrar e remover (“desregistrar”) os escutadores de eventos (*listeners*) que monitoram os dados vindos dos sensores.

Esta API inclui as seguintes classes e interfaces:

2.3.5.1 SensorManager

Esta classe é um serviço do sistema Android que contém vários métodos para acessar e listar sensores presentes em um dispositivo, registrar e remover os *listeners* e processar dados. *SensorManager.getOrientation()* é um exemplo de um método que usa dados do sensor para gerar informações de orientação de um aparelho.

2.3.5.2 Sensor

Esta classe oferece informações a respeito de um sensor específico, tais como:

- Alcance máximo (*maximum range*);
- Tempo de atraso mínimo (*minimum delay*);
- Nome (*name*);
- Consumo de energia em mA (*power*);
- Resolução (*resolution*);
- Tipo (*type*);
- Vendedor (*vendor*);
- Versão (*version*).

2.3.5.3 SensorManager

Esta classe contém dois métodos para acessar os objetos da classe `Sensor`: `getSensorList()` – método que retorna todos os sensores de um determinado tipo e `getDefaultSensor()` que retorna o sensor padrão (*default*) para um tipo específico. É necessário que uma aplicação verifique se o(s) sensor(es) que ela necessita está disponível em um aparelho, porque ele pode ou não contê-lo.

2.3.5.4 SensorEventListener

O *SensorEventListener* é uma interface utilizada para monitorar um sensor. É usada para criar dois métodos que recebem notificações (eventos) quando o valor vindo do sensor correspondente se altera (*onSensorChanged*) ou quando a precisão do mesmo sofre alterações (*onAccuracyChanged*).

2.3.5.5 SensorEvent

O *SensorEvent* é uma estrutura de dados que contém informações passadas para um aplicativo quando determinado sensor possui dados para reportar. Um objeto de *SensorEvent* é passado para os métodos no *SensorEventListener* e é composto por:

- *SensorEvent.accuracy*: Cada sensor reporta uma das quatro exatidões existentes (neste caso, a exatidão se refere ao quão confiável os valores reportados são):

`SensorManager.SENSOR_STATUS_ACCURACY_HIGH`: indica que o sensor está reportando a mais alta exatidão possível;

`SensorManager.SENSOR_STATUS_ACCURACY_MEDIUM`: indica que o sensor está operando com exatidão média, e que a calibração pode melhorar os resultados reportados;

`SensorManager.SENSOR_STATUS_ACCURACY_LOW`: indica que o sensor está reportando a exatidão mais baixa possível e precisa ser calibrado;

`SensorManager.SENSOR_STATUS_ACCURACY_UNRELIABLE`: indica que os dados do sensor não são confiáveis. Pode indicar que é necessário efetuar uma calibração ou que as leituras não são possíveis de serem realizadas no momento. Sensores binários como os sensores de proximidade, por exemplo, sempre reportam esse valor.

- *SensorEvent.sensor*: referência para o sensor que criou o *SensorEvent*;
- *SensorEvent.timestamp*: o tempo em milissegundos que o evento (*SensorEvent*) ocorreu;
- *SensorEvent.values*: contém um vetor com os dados vindos do sensor. O tamanho e o seu respectivo significado variam de acordo com o sensor que o produziu.

2.3.5.6 Taxa de operação dos sensores

Quando um *listener* é registrado, define-se uma taxa de operação para o mesmo. As taxas pré-definidas são:

- `SENSOR_DELAY_FASTEST`;
- `SENSOR_DELAY_GAME`;
- `SENSOR_DELAY_UI` (adequado para funções de interface de usuário, como girar a tela, orientação);
- `SENSOR_DELAY_NORMAL`.

No Android 4.0.3 esses valores correspondem a 0, 20, 67 e 200 milissegundos, respectivamente. O desenvolvedor pode definir qualquer valor para essa taxa. No entanto, os eventos podem ser recebidos mais rápidos ou mais lentos do que o atraso especificado.

Como os valores vindos dos sensores não são necessariamente espaçados igualmente no tempo, o campo *SensorEvent.timestamp()* permite ao desenvolvedor acessar a taxa de operação associada ao dado (o qual é mantido pelo campo *SensorEvent.values*), em nanosegundos. Para encontrar o tempo mínimo de espera entre dois eventos, utiliza-se o método *Sensor.getMinDelay()*.

2.3.6 Descrição dos sensores internos

Para descrever o funcionamento e características dos sensores, foram consultadas as obras de Meier (2012) e Milete e Stroud (2012).

2.3.6.1 Sensores Ambientais

2.3.6.1.1 Sensor de luminosidade (*Sensor.TYPE_LIGHT*)

Geralmente utilizado para ajustar o brilho da tela do aparelho conforme a luz ambiente. O sensor de luz está localizado, normalmente, no visor do dispositivo sob uma pequena abertura de coloração preta *sobre* o vidro. Trata-se de um fotodiodo, que gera uma tensão quando a luz incide sobre o sensor. Apresenta valores em lux e opera numa faixa que varia de 1 a 30.000 lux, com resolução de 1 lux.

Lux (lx) é a unidade de iluminação do Sistema Internacional de Medidas (SI), e mede a incidência perpendicular de 1 lúmen (unidade de medida de fluxo luminoso) em uma superfície de 1 metro quadrado.

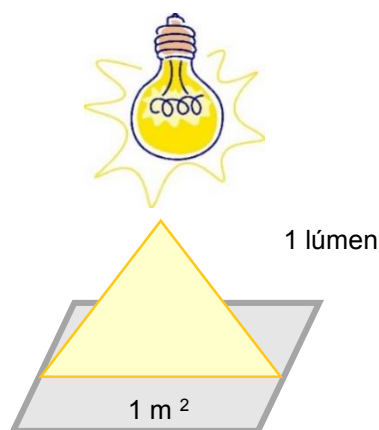


Figura 15 - Unidade de iluminação lux (lx).

FONTE: Elaborada pela autora, com base na imagem extraída do site EAV (Engenharia Visual)⁹

Diferentes condições de iluminação geram valores distintos de luminância. O valor de 0,25 lux, por exemplo, é como o brilho indireto da lua cheia - brilhante o suficiente para ser captado pelos olhos humanos, mas uma câmera sem flash não seria capaz de capturar a luz suficiente para tirar uma foto. Um dia nublado equivale a 10,000 lux; a luz do dia possui valor em torno de 20,000 lux; e a luz solar direta equivale a cerca de 110,000 lux. No entanto, esses números podem variar dependendo do ambiente, devido a presença de sombras e outras fontes de luz, e assim gerar diferentes leituras de lux em determinados locais.

Como os valores obtidos do sensor de luz podem variar conforme o ambiente, são definidas as seguintes constantes (em lux) para facilitar a utilização deste sensor:

- `SensorManager.LIGHT_NO_MOON`: 0,001
- `SensorManager.LIGHT_FULLMOON`: 0,25
- `SensorManager.LIGHT_CLOUDY`: 100
- `SensorManager.LIGHT_SUNRISE`: 400
- `SensorManager.LIGHT_OVERCAST`: 10.000
- `SensorManager.LIGHT_shade`: 20.000
- `SensorManager.LIGHT_SUNLIGHT`: 110.000
- `SensorManager.LIGHT_SUNLIGHT_MAX`: 120.000

⁹ Disponível em < <http://www.eav.eng.br/tech/fisica/NotesImages/Topic14NotesImage4.jpg>>. Acesso em: 10 abr. 2013

2.3.6.1.2 Sensor de proximidade (*Sensor.TYPE_PROXIMITY*)

O sensor de proximidade, geralmente visível apenas à luz do sol, está normalmente localizado na parte superior do aparelho. É constituído por um LED infravermelho fraco (diodo emissor de luz) ao lado de um fotorreceptor.

Um feixe de luz infravermelha é projetado pelo LED e ao ser refletido por algum obstáculo (por exemplo, a orelha de uma pessoa que está realizando uma chamada telefônica), é detectado pelo fotorreceptor. A medida que o obstáculo se aproxima do conjunto emissor-receptor, maior será a intensidade do sinal recebido. Quando o sinal alcançar um limite máximo pré-definido, o sistema detecta a presença deste obstáculo. A Figura 16 representa o funcionamento do sensor:

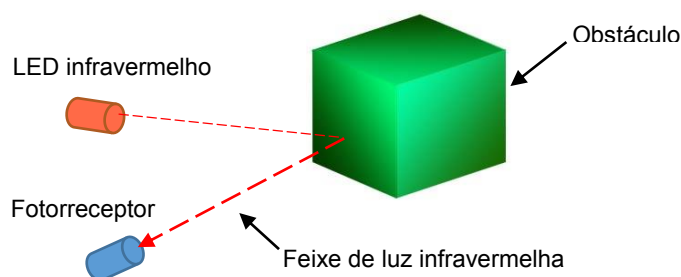


Figura 16- Funcionamento do sensor de proximidade.

FONTE: Elaborada pela autora, com base na imagem disponível no site Mecatrônica Atual.¹⁰

Alguns sensores de proximidade reportam a distância de um objeto em centímetros. Outros retornam valores binários que indicam a proximidade de dado objeto (perto ou longe) – este é o tipo de sensor utilizado nos *smartphones*. Nesses casos, o valor máximo (obtido pelo método *getMaximumRange()*) representa o estado “longe”. Geralmente este valor está na faixa de 5 cm, mas pode variar conforme o sensor.

A principal função do sensor de proximidade é detectar a aproximação do rosto do usuário durante o recebimento de uma ligação telefônica, afim de bloquear a tela para economizar bateria, iniciar algum comando de voz ou prevenir alguns comandos indesejadas que poderiam ocorrer (como cancelar a ligação, por exemplo).

¹⁰ Disponível em < http://www.mecatronicaatual.com.br/files/image/MF20_pg30_figura01.png>. Acesso em abril. 2013

Alguns dispositivos podem não conter este tipo de sensor. Assim, muitos desenvolvedores acabam substituindo o seu uso pelo sensor de luz para bloquear a tela durante as chamadas.

2.3.6.1.3 Sensor de umidade relativa (*Sensor.TYPE_RELATIVE_HUMIDITY*)

O sensor de umidade relativa fornece a umidade ambiente como sendo a porcentagem do vapor de água presente no ar. Mais especificamente, representa a umidade relativa da quantidade máxima de vapor de água que o ar pode conter a uma certa temperatura. Um valor de 100% indica que o ar está completamente saturado, o que favorece a ocorrência de chuvas. O valor retornado por este sensor é o que normalmente as previsões meteorológicas utilizam.

Este sensor pode ser usado juntamente com a temperatura ambiente para calcular o ponto de orvalho e a umidade absoluta. O ponto de orvalho é a temperatura pela qual o vapor da água condensa. Umidade absoluta é a massa da água em um dado volume de ar.

2.3.6.1.4 Sensor de pressão (*Sensor.TYPE_PRESSURE*)

O sensor de pressão atmosférica, ou barômetro, é um dispositivo micro eletromecânico (*Microelectromechanical systems – MEMS*), que retorna a pressão atmosférica em milibars (mbars). Poucos dispositivos contêm esse tipo de sensor.

Normalmente, é utilizado para determinar a altitude em lugares onde dispositivo não pode obter a localização retornada por um GPS, como por exemplo dentro de um prédio, através do método *getAltitude()*. Pode ser utilizado também para mensurar as alterações da pressão atmosférica local em determinado ambiente.

2.3.6.1.5 Sensor de temperatura (*Sensor.TYPE_AMBIENT_TEMPERATURE* e *Sensor.TYPE_TEMPERATURE*)

- *Sensor de temperatura ambiente (Sensor.TYPE_AMBIENT_TEMPERATURE)*: Este sensor retorna a temperatura, em graus Celsius, do ambiente onde se encontra o dispositivo.

- Sensor de temperatura (*Sensor.TYPE_TEMPERATURE*): Este sensor foi projetado para retornar a temperatura, em graus Celsius, da CPU dos dispositivos Android afim de prover a calibração interna do hardware. Como não permite mensurar a temperatura ambiente, costuma não ser útil para os desenvolvedores. Assim, a partir da versão 4.0, foi substituído pelo sensor de temperatura ambiente.

2.3.6.2 Sensores de Orientação e Movimento

Sistema de Coordenadas

Quando os sensores de orientação e movimento são utilizados, dois sistemas de coordenadas são definidos:

- Sistema de coordenadas global: x_e, y_e, z_e ; e
- Sistema de coordenadas do dispositivo: x, y e z

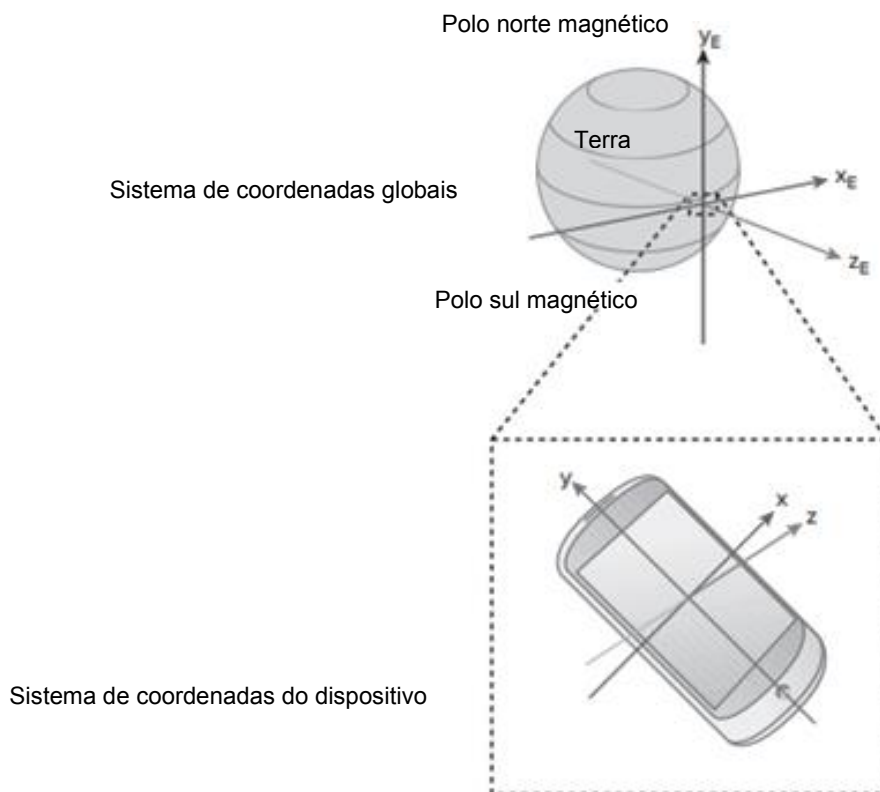


Figura 17 - Sistema de coordenadas Android.

FONTE: Adaptação da imagem disponível em MILETTE; STROUD (2012, p. 90).

Esta figura mostra o dispositivo posicionado na linha do Equador, com uma certa inclinação em relação à Terra. Todos os sistemas de coordenadas para sensores de três eixos devem obedecer a esse sistema, com exceção do sensor de orientação (*Sensor.TYPE_ORIENTATION*) que se tornou obsoleto.

Sistema de coordenadas global

Todos os sistemas e métodos que fazem referência a uma orientação absoluta em relação à Terra, fazem uso do sistema de coordenadas global. Estes incluem:

- O vetor de rotação do sensor, que utiliza o acelerômetro, magnetômetro, e possivelmente o giroscópio para determinar o posicionamento do dispositivo em relação à Terra;
- *getRotationMatrix()*, *getRotationMatrixFromVector()*, e *getQuaternionFromVector()*, que retornam a matriz de rotação ou quatérnion responsáveis por mapear as coordenadas do aparelho em coordenadas do sistema global.
- *getOrientation()* que utiliza a matriz de rotação gerada pelo método *getRotationMatrix()* e retorna um vetor de orientação
- *getInclination()* que utiliza a matriz de rotação do método *getOrientation()* e retorna a inclinação magnética (ângulo obtido pela inclinação da agulha da bússola em relação a um plano horizontal à superfície da Terra).

Neste sistema de coordenadas:

- y_e : aponta para o norte magnético, que é aproximadamente o norte verdadeiro;
- x_e : aproximadamente à leste – paralelo à superfície da Terra mas a 90 graus de y_e
- z_e : aponta sempre para o centro da Terra

Sistema de coordenadas do dispositivo

Os sensores de orientação e movimentação (acelerômetro, magnetômetro e giroscópio) retornam valores correspondentes ao sistema de coordenadas do aparelho. Este sistema é parcialmente definido pela orientação padrão, que difere dependendo do tipo de dispositivo. Por exemplo, a orientação padrão para *smartphones* é retrato, enquanto que para *tablets* é paisagem. Nesse caso, os eixos são direcionados da seguinte maneira:

- eixo x horizontal com valores positivos à direita;
- eixo y vertical com valores positivos acima;
- eixo z com valores positivos à frente da tela.

Este sistema de coordenadas se mantém fixo ao aparelho, ou seja, as orientações dos eixos não sofrem modificações quando o dispositivo altera do modo retrato para paisagem.

Ângulos

A angulação em torno dos eixos é dada por um vetor de três posições, matriz de rotação ou quatérnion (maneira alternativa para a representação das matrizes de rotação) que mapeia o sistema de coordenadas do aparelho para coordenadas globais. Por exemplo, na leitura dos valores obtidos pelo giroscópio, os valores (0.1, -0.2, 0.0) indicam que a taxa de rotação é +0.1 radianos por segundo em torno do eixo x , -0.2 em torno do eixo y e não há rotação em torno do eixo z .

A direção angular é determinada pela chamada Regra da Mão Direita: se o polegar da mão direita apontar ao longo da direção positiva de um eixo, seus dedos irão se curvar na direção do ângulo positivo. Suas componentes podem ser chamadas de azimuth, inclinação e giro, e serão detalhadas mais adiante.

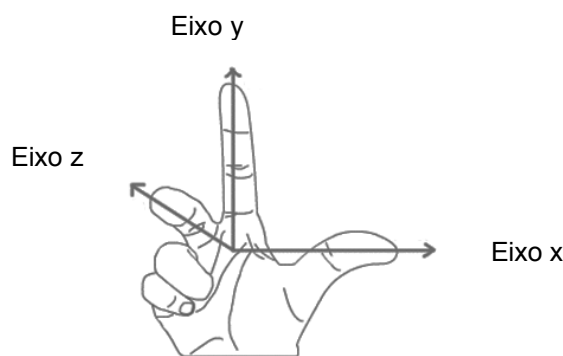


Figura 18 - Regra da mão direita

FONTE: Adaptação da imagem extraída do site LeedsBlog.¹¹

2.3.6.2.1 Acelerômetro, sensor de gravidade e sensor de aceleração linear (*Sensor.TYPE_ACCELEROMETER*, *Sensor.TYPE_GRAVITY*, e *Sensor.TYPE_LINEAR_ACCELERATION*)

O acelerômetro é um dispositivo micro eletromecânico (*Microelectromechanical systems – MEMS*) utilizado para medir a aceleração de um objeto em relação à gravidade.

O princípio de funcionamento deste sensor é o de sistema e mola e baseia-se nas:

- Lei da Inércia: quando um objeto sofre uma aceleração, o peso, por efeito da inércia, tende a conservar a sua velocidade, deslocando-se em relação ao eixo. A magnitude deste deslocamento dá uma medida da aceleração. (INFOPÉDIA, 2003).

- Lei de Hooke: quando uma mola sofre uma deformação, ou seja, uma variação $x = |L - L_0|$ no seu comprimento, seja para mais (alongamento da mola) ou para menos (compressão da mola), a mola reage aplicando uma força restauradora F que tenta levar a mola de volta à situação inicial e cuja intensidade vale $F = k \cdot x$ (1) (Figura 19). K é uma constante típica da mola e que caracteriza a sua elasticidade, ou seja, é maior quanto mais dura (ou mais difícil de deformar) for a mola e vice-versa. (JÚNIOR, 2008).

¹¹ Disponível em < <https://elgg.leeds.ac.uk/bms4a2r/weblog/10935.html>>. Acesso em abril. 2013

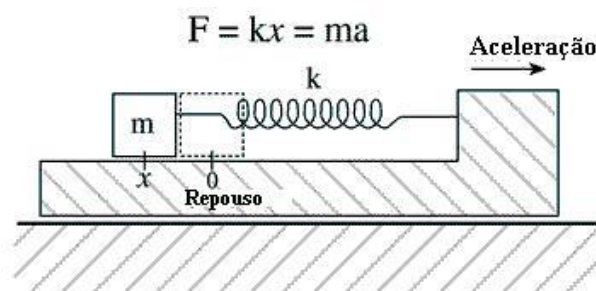


Figura 19 - Princípio da Lei de Hooke.
 FONTE: (JÚNIOR, 2008).

• Segunda Lei de Newton: onde força, massa e aceleração estão relacionadas através da equação $F = m \cdot a$ (2). Assim, a aceleração adquirida por um dado objeto ou corpo é diretamente proporcional à força que sobre ele atuam. Igualando a equação (1) com (2), tem-se que:

$$m \cdot a = k \cdot x$$

$$a = \frac{k \cdot x}{m} \quad (3)$$

Desta maneira, a aceleração pode ser dada pela equação (3). A Figura 20 apresenta os princípios físicos descritos acima. No entanto, vale ressaltar que este sistema exemplifica um acelerômetro de um único eixo. Caso o sensor meça a aceleração nos três eixos (x, y e z) é necessário um sistema deste para cada um dos eixos.

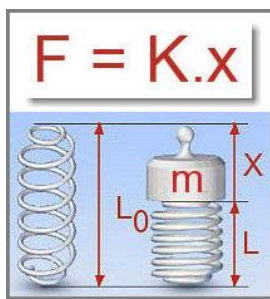


Figura 20 - Sistema de massa usado para medir aceleração
 FONTE: (BUDZINSKI, 1999)

Assim sendo, a aceleração provoca uma deformação na mola que permite determinar a força aplicada a determinado objeto. Na Figura 21, por exemplo, A representaria um dispositivo em repouso sobre a mesa; B representaria o dispositivo no momento em que o usuário movimenta o mesmo para a direita; e C corresponderia

a um dispositivo em queda livre (neste caso a aceleração reportada pelo sensor seria 0 m/s^2 , uma vez que as molas não iriam sofrer deformação).

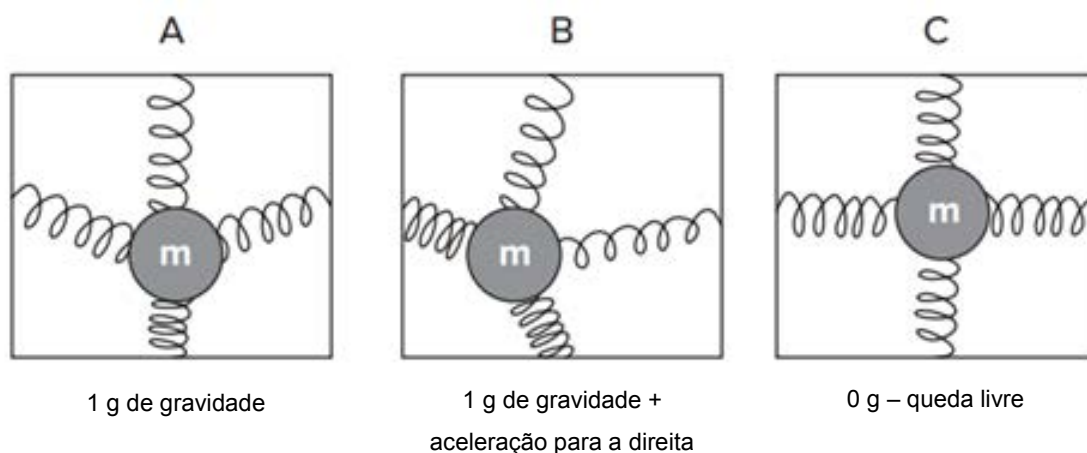


Figura 21 - Força aplicada a uma massa suspensa por molas (g é a unidade de aceleração e equivale a 9.8 m/s^2)

FONTE: Adaptação da imagem disponível em MILETTE; STROUD (2012, p. 92).

A partir do Android 2.3 os sensores sintéticos de gravidade (`Sensor.TYPE_GRAVITY`) e aceleração linear (`Sensor.TYPE_LINEAR_ACCELERATION`) se tornaram disponíveis. A soma dos valores destes sensores equivale ao valor do acelerômetro.

Unidades do sensor e resolução

O Android retorna a aceleração em m/s^2 . A aceleração da gravidade na Terra é de $9,8 \text{ m/s}^2$ (dada pelas constantes `SensorManager.GRAVITY_EARTH` e `SensorManager.STANDARD_GRAVITY`, ambas equivalentes a $9,80665 \text{ m/s}^2$) ou $1g$. Normalmente o sensor opera em uma taxa de 0 ± 2 ou $\pm 4g$, com resolução de $0,1 \text{ m/s}^2$. Quando em repouso o sensor retorna o valor de $+9,8 \text{ m/s}^2$ para o eixo z. Na maioria dos casos, a convenção adotada é:

- `values[0]` : - g_x (g_x negativo) no eixo x
- `values[1]` : - g_y (g_y negativo) no eixo y
- `values[2]` : - g_z (g_z negativo) no eixo z

onde g_x, g_y e g_z representam as posições do vetor de aceleração.

2.3.6.2.2 Giroscópio (Sensor.TYPE_GYROSCOPE)

Os giroscópio tem seu funcionamento baseado nos mesmos princípios do acelerômetro, porém ao invés de medir a aceleração, foi projetado para medir a chamada força de Coriolis, provocada pela rotação de um objeto.

Segundo Menezes ([200-?]), o efeito Coriolis é “a tendência que qualquer corpo em movimento sobre a superfície terrestre tem de mudar seu curso devido à direção rotacional e da velocidade da Terra.” Como exemplo, considere uma pessoa sentada em uma plataforma rotativa. Ao jogar uma bola para a frente, a mesma parece curvar na plataforma como se uma força estivesse atuando sobre ela. No entanto, para uma pessoa que acompanha a cena próxima à plataforma, a bola parece seguir a trajetória em linha reta (como esperado conforme as Leis de Newton).

Estes dispositivos MEMS são compostos por uma massa rotatória fixa e tem as duas pontas do eixo fixadas em um suporte giratório, conforme ilustrado na Figura 22. Quando o sensor é movimentado, a força de Coriolis faz com que essa massa se desvie em direção ao movimento realizado, e então o giroscópio começa a se mover ao longo de um eixo diferente. O movimento ao longo desse novo eixo é detectado eletricamente por capacitores – um deles está fixado à estrutura e outro à massa.



Figura 22 - Representação de um giroscópio
FONTE: Imagem extraída do site Wikipédia.¹²

A força de Coriolis atua somente quando o dispositivo está rotacionando. Assim, o giroscópio retorna apenas a velocidade angular ou a velocidade de rotação.

¹² Disponível em < http://en.wikipedia.org/wiki/File:Gyroscope_operation.gif>. Acesso em: 10 abril. 2013

Quando o dispositivo estiver em repouso, o sensor retornará 0 para os três eixos (x,y e z).

Unidades do sensor

O Android retorna valores em radianos por segundo ao longo dos três eixos x, y e z mostrados na Figura 17. O padrão matemático seguido é: se o eixo em questão está apontando para sua direção, a rotação anti-horária indica valores positivos, conforme a regra da mão direita descrita anteriormente.

Taxa de operação

A taxa típica máxima esperada é de cerca de 35 graus/segundo (0.61 rad/s) e resolução de cerca de 0.001 graus/segundo (2E-5 rad/s).

2.3.6.2.3 Sensor de campo magnético (Sensor.TYPE_MAGNETIC_FIELD)

Sensores de campo magnético podem operar sob vários métodos diferentes, dependendo do fabricante e arquitetura projetada. Podem usar o efeito Hall, materiais magneto-resistivos ou a força de Lorentz (força magnética que atua sobre uma carga em movimento). (SHOW DE FÍSICA, 2008). Mas, independentemente do mecanismo físico empregado estes sensores irão retornar o campo magnético nos três eixos (x,y e z) – cada eixo é composto por um sensor deste.

Os sensores mais populares no mercado são os que utilizam o efeito Hall. De acordo com Teleco (2013), este “consiste no fenômeno segundo o qual, perante a presença de um campo magnético dirigido perpendicularmente a um condutor, pelo qual circula corrente, aparece uma diferença de potencial nas faces opostas a circulação de corrente.”

Unidades do sensor, taxa e resolução

O Android retorna o valor do campo magnético em microtesla (μT). A taxa típica do sensor é de cerca de 2000 μT , com resolução de 0,1 μT . No entanto, o valor absoluto retornado pelo sensor não importa muito, pois estes variam de acordo com

o ambiente local. Por exemplo, se um corpo metálico se aproximar do sensor e for removido, a leitura no sensor pode ser alterada.

Se for necessário obter valores mais precisos, a classe *android.hardware.GeomagneticField* pode ser utilizada para estimar a direção e magnitude do campo magnético em um determinado ponto na Terra. Para isso, é preciso informar a latitude, longitude, altitude e tempo para instanciá-la, afim de obter acesso aos seguintes campos:

- *float getDeclination()* – a declinação magnética é o ângulo formado entre a direção do Norte Verdadeiro e ao do Norte Magnético, em determinado local da superfície terrestre (Figura 23). (SISPESCA, s.d).

- *float getFieldStrength();*

- *float getHorizontalStrength();*

- *float getInclination()* – inclinação magnética é o ângulo formado pela bússola com um plano horizontal à superfície da Terra. (VIANA, s.d);

- *float getX(), float getY(), float getZ()* - retornam os valores do campo magnético em nanoteslas.

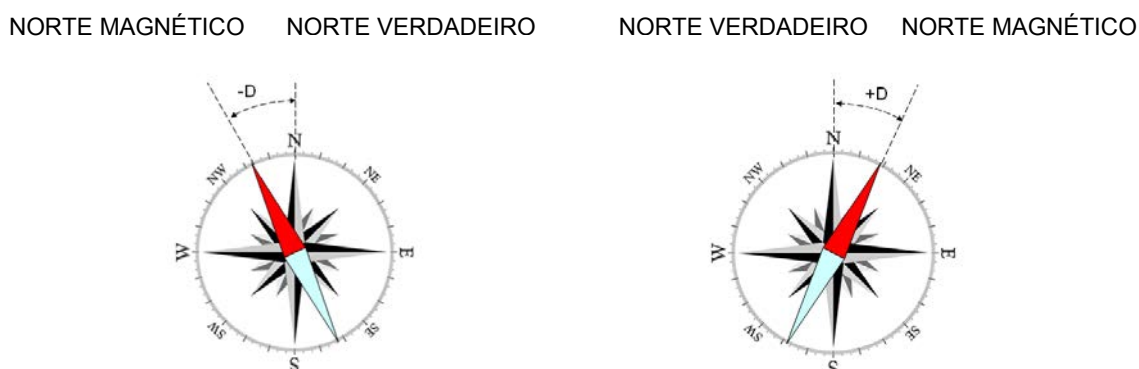


Figura 23 - Declinação magnética a partir da direção do Norte Verdadeiro
 FONTE: Adaptação da imagem extraída do site Heliodon.¹³

¹³ Disponível em < http://www.heliodon.com.br/calcular_declinacao_2.html>. Acesso em: 10 abril. 2013

O Android associa as seguintes constantes para o sensor `Sensor.TYPE_MAGNETIC_FIELD` com seus respectivos valores (em microtesla):

- `SensorManager.MAGNETIC_FIELD_EARTH_MAX` : 60.0 μT
- `SensorManager.MAGNETIC_FIELD_EARTH_MIN` : 30.0 μT

2.3.6.2.4 Vetor de rotação (*Sensor.TYPE_ROTATION_VECTOR*)

Disponível a partir da API nível 9, o *Sensor.TYPE_ROTATION_VECTOR* calcula o ângulo de rotação do sistema de coordenadas globais com relação ao sistema de coordenadas do aparelho, usando o acelerômetro, magnetômetro, e possivelmente o giroscópio se disponível.

A saída do sensor apresenta valores similares a um quatérnion, que é uma representação alternativa de uma rotação. Quatérnions têm certas vantagens matemáticas em relação às coordenadas euclidianas, mas não são facilmente visualizados. Assim, a representação Euclidiana é normalmente a mais utilizada e assim para isso

A saída do sensor é convertida para uma matriz de rotação utilizando o método *getRotationMatrixFromVector()*, que recebe dos argumentos. O primeiro é uma matriz de 9 ou 16 elementos que contém a matriz de rotação desejada (descreve a rotação necessária para transformar o sistema de coordenadas globais para o sistema de coordenadas do aparelho). O segundo é a saída do vetor de rotação retornado pelo sensor.

SensorManager.getOrientation()

A matriz de rotação pode ser obtida através dos métodos *getRotationMatrix()* ou *getRotationMatrixFromVector()*, e então ser passada para *getOrientation()* para obter a direção de orientação, em radianos. O vetor obtido representa a orientação do aparelho em relação à Terra (Figura 23). Este, difere da orientação definida para a tela de um dispositivo (retrato ou paisagem) definido pelo campo *getResources ().getConfiguration ().orientation*.

O sistema de coordenadas utilizado é mostrado na Figura 24. Os eixos x e z são invertidos com relação ao sistema de coordenadas globais mostrado na Figura

17. Os valores reportados são todos dados pela regra da mão direita. Apesar do `Sensor.TYPE_ORIENTATION`, que retorna valores em graus, ter se tornado obsoleto, continua sendo amplamente implementado pelos desenvolvedores.

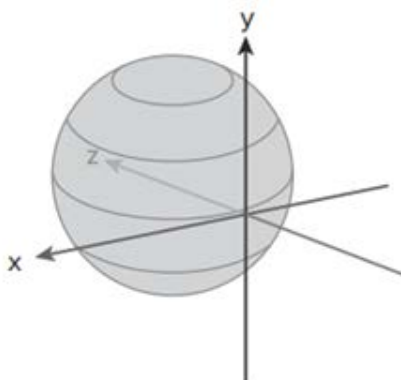


Figura 24 - Os eixos referentes a `getOrientation()`.¹⁴

FONTE: Adaptação da imagem disponível em MILETTE; STROUD (2012, p. 90).

getOrientation() retorna os seguintes valores:

- `values[0]` : Azimute (graus de rotação em torno do eixo z): é o ângulo formado entre o norte magnético e o eixo y do dispositivo. Assume os seguintes valores: 0, se o eixo y estiver alinhado com o norte magnético; 180 se estiver apontando para o sul; 90 para leste; e 270 para oeste.
- `values[1]` : Inclinação (graus de rotação em torno do eixo x): apresenta valores positivos quando o eixo z positivo é rotacionado na direção do eixo y positivo; assume valores negativos quando o eixo z positivo rotacionar na direção do eixo y negativo. A faixa de valores pode variar de 180 a -180 graus.
- `values[2]` : Giro (graus de rotação em torno do eixo y): apresenta valores positivos quando o eixo z positivo for rotacionado na direção do eixo x positivo e negativo quando o eixo z positivo girar na direção do eixo x negativo. A faixa de valores varia de 90 a -90 graus.

¹⁴ A esfera representa a Terra e o dispositivo está posicionado na linha do Equador. O eixo y aponta sempre para o norte magnético, x para oeste, e z para o centro da Terra. (X e z são invertidos com relação a Figura 17).

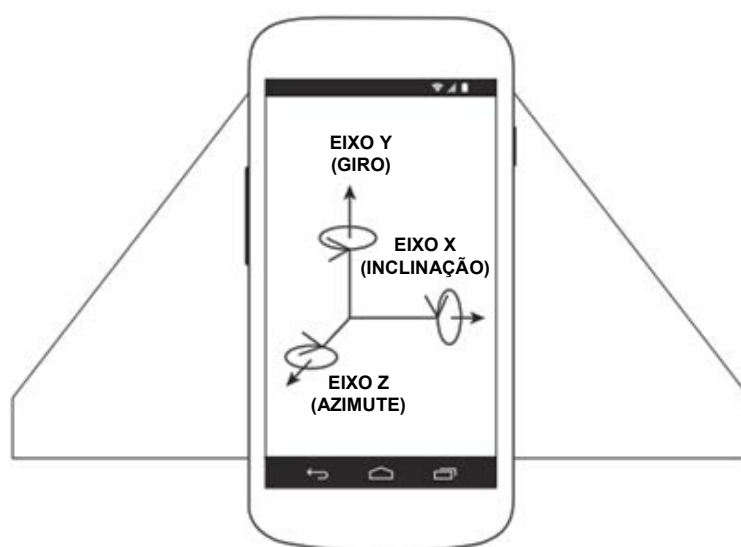


Figura 25 - Representação dos valores retornados pelo método `getOrientation()`.
 FONTE: Adaptação da imagem disponível em MEIER, 2012, p. 493.

2.3.7 Interpretando os valores dos sensores

As seções anteriores apresentaram informações referentes ao modo de funcionamento e utilização dos sensores internos da plataforma Android. Como visto, os valores retornados por cada um deles pode variar dependendo do tipo de sensor monitorado (Quadro 4).

Quadro 4 - Valores retornados pelos sensores disponíveis na plataforma Android

Categoria do Sensor	Sensor	Valores retornados	Unidades da medida	Descrição dos valores retornados
Sensores Ambientais	TYPE_AMBIENT_TEMPERATURE	value[0]	graus Celsius (°C)	Temperatura ambiente
	TYPE_LIGHT	value[0]	lux (lx)	Iluminação ambiente
	TYPE_PRESSURE	value[0]	milibars (mbars)	Pressão atmosférica
	TYPE_RELATIVE_HUMIDITY	value[0]	%	Umidade relativa do ambiente

	TYPE_TEMPERATURE	value[0]	graus Celsius (°C)	Temperatura interna do aparelho
Sensores de Movimento	TYPE_ACCELEROMETER	value[0]: eixo x value[1]: eixo y value[2]: eixo z	m/s ²	Aceleração ao longo dos três eixos (x,y e z), considerando a força da gravidade
	TYPE_GRAVITY	value[0]: eixo x value[1]: eixo y value[2]: eixo z	m/s ²	Força da gravidade ao longo dos três eixos (x,y e z)
Sensores de Movimento	TYPE_GYROSCOPE	value[0]: eixo x value[1]: eixo y value[2]: eixo z	radianos/segundos (r/s)	Taxa de rotação em torno do três eixos
	TYPE_LINEAR_ACCELERATION	value[0]: eixo x value[1]: eixo y value[2]: eixo z	m/s ²	Aceleração ao longo dos três eixos (x,y e z), não levando em consideração a força da gravidade
	TYPE_ROTATION_VECTOR	values[0]: $x \cdot \sin(\theta/2)$	ângulo (°)	Orientação do dispositivo descrita como um ângulo de rotação

		values[1]: $y \cdot \sin(\theta/2)$ values[2]: $z \cdot \sin(\theta/2)$ values[3]: $\cos(\theta/2)$ (opcional)		em torno de um eixo.
Sensores de Orientação	TYPE_MAGNETIC_FIELD	value[0]: eixo x value[1]: eixo y value[2]: eixo z	microtesla (μT)	Campo magnético ambiente
Sensores de Orientação	TYPE_ORIENTATION	value[0]: azimuth value[1]: inclinação value[2]: giro	graus	Graus de rotação que o aparelho faz em torno dos três eixos (x, y e z).
	TYPE_PROXIMITY	value[0]	cm	Distância de um objeto

FONTE: Adaptação dos quadros presentes em Android Developers, 2013a, 2013b e 2013c.

2.3.8 Erros e processamento dos valores vindos dos sensores internos

Os valores obtidos dos sensores podem, em alguns casos, representar dados incorretos devido à ruídos ou em razão da degradação do sensor que ocorre ao longo do tempo.

Para resolver esse problema, se faz necessário o uso de alguns algoritmos e técnicas capazes de lidar com tais erros. Por exemplo, pode-se reduzir a presença de

erros através da filtragem dos valores obtidos ou pela combinação de vários valores obtidos por distintos sensores.

Nos tópicos abaixo serão estudados métodos para tratar erros provenientes dos sensores internos da plataforma Android.

2.3.8.1 Terminologias utilizadas para descrever erros

As definições aqui apresentadas são importantes para compreender diferentes terminologias utilizadas para descrever erros, quais tipos de erros podem ocorrer e quais são as técnicas existentes para tratá-los.

2.3.8.1.1 *Exatidão e precisão*

Para avaliar a qualidade da medição obtida pelo sensor são necessários dois conceitos importantes:

- *Exatidão*: indica o grau de aproximação entre as medidas obtidas do valor “verdadeiro” (normalmente aceito como referência) ou real. Pode ser descrita em termos de erro relativo ou absoluto.
- *Precisão*: indica o grau de aproximação entre medidas que foram obtidas experimentalmente da mesma forma. Uma medida mais precisa é aquela que contém mais algarismos significativos.

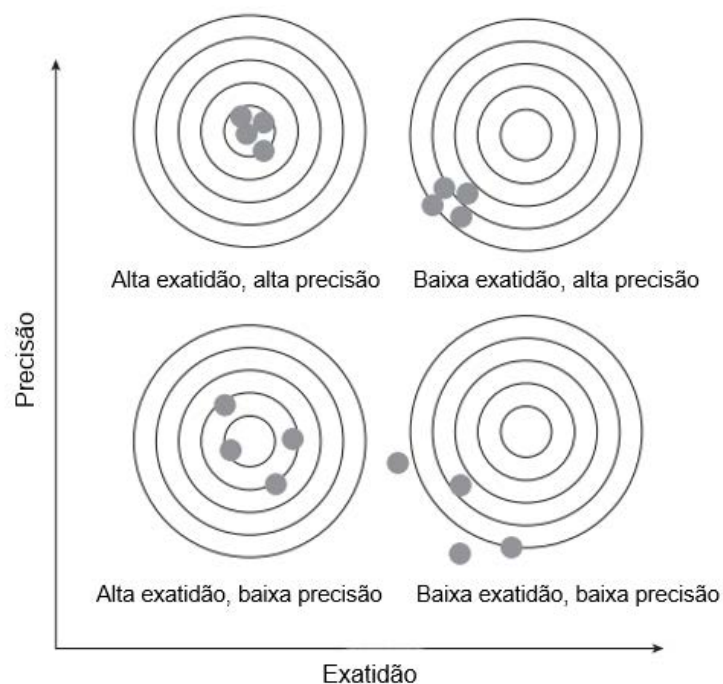


Figura 26 - Exatidão versus precisão

FONTE: Adaptação da imagem disponível em MILETTE; STROUD (2012, p. 104).

2.3.8.1.2 Tipos de erros

Erro humano, sistemático, e aleatório

- Erros humanos: são erros ocasionados por humanos ao realizar uma medição (tais como leitura incorreta de um valor representado em um gráfico, por exemplo);
- Erros sistemáticos: são erros que afetam o valor de uma medida, mas geralmente não são fáceis de identificar. (CRUZ, s.d). Ocorrem quando o método empregado na medição falha ou devido a utilização de um instrumento com defeito ou mal calibrado. Quando identificados, podem ser removidos do sistema;
- Erros aleatórios: são inerentes à medição (tais como ruídos, variações de condições ambientais) e assim, não podem ser completamente removidos do sistema.

Ruídos

Ruído é a flutuação aleatória da medida de um valor, ocasionado por interferências vindas do ambiente em que os sensores estão inseridos. (PROGRAMA DE ENGENHARIA BIOMÉDICA – UFRJ, [200-?]).

Deriva (drift)

Representa a mudança do valor de saída do sensor com o passar do tempo. O erro devido à deriva pode ocorrer devido ao envelhecimento de componentes, variações de temperatura e umidade, por exemplo.

Erro de Offset (ou “Offset”, ou “Bias”)

Ocorre quando a saída do sinal reportada pelo sensor não é zero para uma medida que deveria ser zero. Por exemplo, se um sensor está sobre a mesa, o valor retornado pelo acelerômetro deveria ser, exatamente, (0,0, -9.80665 m/s²). Caso contrário, o sensor tem um erro de offset. (SAKURAY, s.d).

Atraso e eliminação de dados

Como o Android não é um sistema operacional de tempo real, algumas medições podem sofrer atraso e serem até mesmo descartadas quando o dispositivo está em uso. Normalmente, isso não é uma preocupação para os desenvolvedores, mas é importante quando possíveis erros estão sendo avaliados.

2.3.8.1.3 Técnicas para tratar erros

A seção anterior discutiu quais os tipos de erros podem ser encontrados ao utilizar os dados dos sensores presentes no Android. Como, na maioria das vezes, será necessário lidar com casos em que tais erros estão presentes, a presente seção apresenta métodos para tratá-los.

Zerar: Se um erro de offset estiver afetando uma aplicação, pode ser útil zerar as medições do sensor.

Filtros: A filtragem de dados mensurados por sensores é uma outra técnica que pode ser utilizada para tratar erros. Nos tópicos “Filtros” serão apresentados quais métodos podem ser empregados para esta filtragem.

Fusão de sensores: Consiste em empregar vários sensores para realizar determinada medição com a finalidade de obter dados mais precisos.

2.3.9 Filtros

Como discutido na seção anterior, a filtragem dos dados obtidos por um sensor é uma técnica que pode ser utilizada para tratar erros. Nesta seção serão apresentados os tipos de filtros que podem ser empregados para tal propósito.

2.3.9.1 Filtro passa baixa

Embora os sensores encontrados nos dispositivos móveis estejam melhorando continuamente, em alguns aplicativos pode ser necessário encontrar maneiras de “suavizar” (*smoothing*) os dados, também conhecido como filtro passa baixa (porque permite que os sinais com frequência abaixo de determinada frequência passem para a saída, e elimina quaisquer sinais com frequências superiores (VITORINO, s.d).

Se você deseja obter a componente de gravidade do acelerômetro, é preferível utilizar o sensor de gravidade (Sensor.TYPE_GRAVITY), que consiste em um filtro passa baixa do acelerômetro, ao invés de filtrar os dados vindos do acelerômetro.

Suavização ponderada

Um método comum de implementar o filtro passa baixa para “suavizar” os dados consiste em ponderar o novo com o antigo valor vindo do sensor. Um parâmetro de suavização (ou valor ponderado) a é usado:

$$(Novo\ valor) = (Valor\ antigo) + x_i * a - (Valor\ antigo) * a \quad (4)$$

que pode ser reescrito como:

$$(Novo\ valor) = (Valor\ antigo) * (1 - a) + x_i * a \quad (5)$$

onde x_i representa valor mensurado mais recente e a um valor que varia conforme a aplicação, normalmente com valores entre 0 e 1. Em jogos que utilizam o acelerômetro para controlar um personagem por exemplo, 'a' normalmente é definido como 0.1. Se o valor de a está próximo de 1, o novo valor será x_i , e se a está próximo de 0 o novo valor não mudará com o cálculo.

Em Java, poderia ser implementado da seguinte maneira:

```
float a = 0.1f;

public void onSensorChanged(SensorEvent event)
{
    x = event.values[0];
    y = event.values[1];
    z = event.values[2];
    mLowPassX = lowPass(x, mLowPassX);
    mLowPassY = lowPass(y, mLowPassY);
    mLowPassZ = lowPass(z, mLowPassZ);
}

//simple low-pass filter
float lowPass(float current, float last)
{
    return last * (1.0f - a) + current * a;
}
```

Figura 27 - Implementação do filtro passa baixa

FONTE: Implementação disponível em MILETTE; STROUD (2012, p. 108).

Média Móvel Simples

Esse método proporciona uma melhor suavização das flutuações nos dados e é utilizado para mostrar um padrão ou tendência de forma mais clara (MICROSOFT OFFICE, 2013). Consiste em encontrar a média aritmética dos k valores mais recentes numa série de dados, sendo a escolha de $k > 1$ arbitrária. Enquanto esses k valores não forem coletados, a técnica não pode ser aplicada.

Escolhendo um parâmetro de suavização

A constante de tempo do filtro oferece a duração de atuação de um sinal. A constante de tempo definida na equação (5) com coeficiente a e período de amostragem dt é:

$$\tau = \frac{a * dt}{(1 - a)} \Leftrightarrow a = \frac{\tau}{\tau + dt} \quad (6)$$

Se a constante de tempo desejada e a taxa de amostragem são conhecidas, a constante do filtro a pode ser escolhida. Embora as taxas de amostragem possam ser especificadas, o Android não garante que os dados do sensor serão entregues na taxa especificada.

2.3.9.2 Filtro passa alta

Um filtro passa-alta funciona de maneira similar ao passa-baixa, mas de maneira inversa: deixa passar para a saída apenas os sinais com frequências acima de determinado valor. O sensor de aceleração linear (Sensor.TYPE_LINEAR_ACCELERATION) é um exemplo de sensor que utiliza este filtro.

Filtro passa baixa inverso

A maneira mais simples de implementar um filtro passa-alta é utilizar um filtro passa-baixa e subtrair o resultado dos dados do sensor. Por exemplo, para filtrar os dados do acelerômetro pode-se utilizar o código abaixo:

```
public void onSensorChanged(SensorEvent event)
{
    final float alpha = 0.8;

    gravity[0] = a * gravity [0] + (1 - a) * event.values[0];
    gravity[1] = a * gravity [1] + (1 - a) * event.values[1];
    gravity[2] = a * gravity [2] + (1 - a) * event.values[2];

    linear_acceleration[0] = event.values[0] - gravity[0];
    linear_acceleration[1] = event.values[1] - gravity[1];
    linear_acceleration[2] = event.values[2] - gravity[2];
}
```

Figura 28 - Implementação do filtro passa baixa inverso

FONTE: Implementação disponível em MILETTE; STROUD (2012, p. 112).

É necessário começar com uma medição inicial ou descartar quaisquer medidas iniciais que sejam iguais a zero. Uma maneira para resolver esse problema é inicializar os valores com a primeira medição. A implementação de um filtro passa-alta é mostrada abaixo:

```
public void onSensorChanged(SensorEvent event)
{
    x = event.values[0];
    y = event.values[1];
    z = event.values[2];
    mHighPassX = highPass(x, mLastX, mHighPassX);
    mHighPassY = highPass(y, mLastY, mHighPassY);
    mHighPassZ = highPass(z, mLastZ, mHighPassZ);
    mLastX = x;
    mLastY = y;
    mLastZ = z;
}

//simple high-pass filter
float highPass(float current, float last, float filtered)
{
    return a * (filtered + current - last);
}
```

Figura 29 - Implementação de um filtro passa-alta

FONTE: Implementação disponível em MILETTE; STROUD (2012, p. 112).

2.3.9.3 Filtro passa banda

Um filtro de passa banda deixam passar apenas os sinais que estão dentro de uma determinada faixa de frequências e rejeita os demais. (SABER ELETRÔNICA, 2008). No caso das aplicações para a plataforma Android, a forma mais comumente usada consiste na combinação dos dois sensores discutidos anteriormente.

2.4 Fotossíntese

A fotossíntese foi o conceito científico escolhido para ser trabalhado neste projeto. Nos próximos tópicos serão detalhados os conceitos necessários para o desenvolvimento do objeto de aprendizagem proposto, disponíveis em Hall (1980), Laurence (2005) e Taiz e Geiger (2009).

2.4.1 Definição e Importância

A vida na Terra, em última análise, depende da energia proveniente do Sol. A fotossíntese é o único processo de importância biológica capaz de tirar proveito dessa energia em um processo onde as plantas sintetizam compostos orgânicos a partir de matéria-prima inorgânica na presença de luz solar.

O termo fotossíntese significa, literalmente, “síntese que utiliza luz”. Os organismos fotossintéticos usam a energia solar para sintetizar combinações orgânicas (utilizam a energia solar para impulsionar a síntese de carboidratos e a liberação de oxigênio a partir de dióxido de carbono e água) que não podem ser desenvolvidas sem que haja a contribuição de energia. (GALLO, 2011).

Todos os seres vivos dependem direta ou indiretamente desse processo. As plantas são responsáveis por tornar o ar respirável para os animais e o homem, pois lançam oxigênio (O_2) e absorvem gás carbônico (CO_2).

2.4.2 Histórico

O início dos estudos sobre a fotossíntese tem início na Grécia Antiga. O filósofo grego Aristóteles acreditava que o solo era responsável por fornecer todos os nutrientes necessários para o crescimento das plantas.

Em 1684, o médico e alquimista Jan Baptist Van Helmont provou com seu experimento, que o solo sozinho não era suficiente para nutrir as plantas. Nele, Van Helmont cultivou um salgueiro em um vaso de cerâmica, fornecendo apenas água à planta. Após cinco anos, observou que a quantidade de solo no vaso era a mesma, mas a planta apresentou 73kg de ganho de peso (saltou de dois para setenta e cinco quilos). Assim, concluiu que a água era responsável pelo crescimento da planta.

O cientista inglês Stephan Hales sugeriu, em 1727, que para o crescimento das plantas também estavam envolvidos luz e o ar.

Em 1771, o químico Joseph Priestley concluiu que as plantas eram responsáveis por “restaurar” o ar. Em seu primeiro experimento, foi colocado um camundongo em uma campânula de vidro fechada. Passado um tempo, o animal morreu devido ao “esgotamento” de ar. O segundo experimento consistiu em repetir o primeiro, só que dessa vez uma planta foi colocada na campânula, junto ao animal. Neste caso, tanto a planta quanto o animal se mantiveram vivos. Priestley observou

que as plantas, ao contrário dos animais, eliminam oxigênio e deu o primeiro passo importante para o estudo da fotossíntese.

Em 1778, o físico Jan Ingenhousz refez os experimentos de Priestley, confirmando suas teorias, descobrindo também que a “restauração” ou “purificação” do ar ocorria somente na presença da luz. Sugeriu que, na presença da luz solar, a planta consome gás carbônico, elimina oxigênio e armazena o carbono como fonte de alimento. Segundo seus experimentos, apenas as partes verdes das plantas eram responsáveis por “purificar o ar”. No ano de 1779, Ingenhousz documentou este papel essencial da luz na fotossíntese.

No ano de 1800 o cientista alemão Thomas Elgemann projetou um espectro de luz sobre os cloroplastos espirais da alga verde filamentosa *Spirogyra* e observou que bactérias dependentes de oxigênio introduzidas no sistema acumulavam-se na região do espectro onde os pigmentos de clorofila absorviam. Esses espectro de ação forneceu as primeiras indicações sobre a efetividade da luz absorvida pelos pigmentos no processo da fotossíntese.

O cientista Nicholas Theodore de Saussure demonstrou, em 1804, que a água também era uma substância importante para o desenvolvimento da planta. Segundo seus experimentos, durante o processo da fotossíntese ocorrem trocas de volumes iguais de oxigênio e gás carbônico e a planta retém carbono e ganha peso.

Na segunda década do século XIX, descobriu-se que a clorofila era responsável pela coloração verde das plantas, além de desempenhar um papel importante na fotossíntese. Na época, mais precisamente em 1864, o botânico e fisiologista alemão Julius Von Sanchs demonstrou a função da clorofila, localizada nos cloroplastos.

Em 1893, Charles Reid Barnes propôs os termos fotossintaxe e fotossíntese para designar o processo biológico da síntese de compostos de carbono complexos a partir de ácido carbônico na presença de clorofila e sob a influência da luz. Com o tempo, o termo escolhido para uso foi “fotossíntese”.

No final do século XIX, outros cientistas estabeleceram os papéis do gás carbônico (CO_2) e da água (H_2O) e mostraram que a matéria orgânica (os carboidratos e o oxigênio) são produtos da fotossíntese. A reação química geral da fotossíntese, em equilíbrio, podia ser escrita da seguinte forma: $6 CO_2 + 6 H_2O \rightarrow \text{luz, planta} \rightarrow C_6H_{12}O_6 + 6 O_2$.

O físico alemão Max Plank descobriu em 1900 a existência de fótons (partículas elementares provenientes da luz, tais como a luz do Sol, estrelas e lâmpada elétrica,

por exemplo). Propôs, também, que a energia podia ser liberada ou absorvida pelos átomos de “pacotes” denominados por Planck de *quantum*, que significa quantidade fixa. Sendo assim, todo átomo absorve ou emite quantidade múltiplas de um valor fixo. Esse valor é conhecido como Constante de Planck (h), e sua unidade é joule.segundos.

Ao estudar os efeitos da temperatura, concentração de CO_2 e a intensidade luminosa sobre a fotossíntese, o fisiologista Frederick Frost Blackman concluiu, em 1905, que este processo consistia em dois tipos de reações: as que dependiam da luz (reações fotoquímicas) e aquelas que ocorriam no escuro (reações bioquímicas).

Durante a década de 1920, o pesquisador Van Niel realizou estudos com bactérias fotossintetizantes, e formulou a teoria de que ao invés do dióxido de carbono, era a água que se degradava gerando oxigênio na fotossíntese. Niel concluiu que a fotossíntese é um processo redox. Segundo o ICNT Redoxoma (2013), “processos redox estão relacionados a reações químicas nas quais há mudança do estado de oxidação dos átomos (reações de óxido-redução/ reações redox)”. Essa conclusão tem servido como um conceito fundamental no qual se basearam todas as pesquisas subsequentes sobre fotossíntese.

No ano de 1940 a 1953, o pesquisador Calvin e seus colaboradores confirmaram as conclusões de Van Niel e a partir de outros experimentos, identificaram o papel do carbono na fotossíntese.

Em 1950 o Ciclo de Calvin foi elucidado em uma série de elegantes experimentos realizados por Calvin, Benson, Bassham e seus colegas. Calvin recebeu o prêmio Nobel em 1961 por esse trabalho. No Ciclo de Calvin, CO_2 e água do ambiente são combinados enzimaticamente com uma molécula aceptora contendo cinco átomos de carbono para gerar duas moléculas de um intermediário com três carbonos.

Em 1954, Daniel Israel Arnon e seus colaboradores conseguiram obter cloroplastos a partir das folhas de espinafre e reproduziram em laboratório as reações completas do processo de fotossíntese.

Mais recentemente, em 2010, um estudo na Universidade de Tel Aviv descobriu que a Mangava-oriental (*Vespa orientalis*) converte luz solar em energia elétrica usando um pigmento chamado *xanthopteri*. Essa é a primeira evidência científica de um membro do Reino Animal fazendo fotossíntese.

2.4.3 O processo da fotossíntese

O órgão da planta responsável por realizar a fotossíntese é a folha, pois possui em suas células dois parênquimas clorofilianos com grande quantidade de cloroplasto (é no interior dessa estrutura que ocorre a transformação de energias).

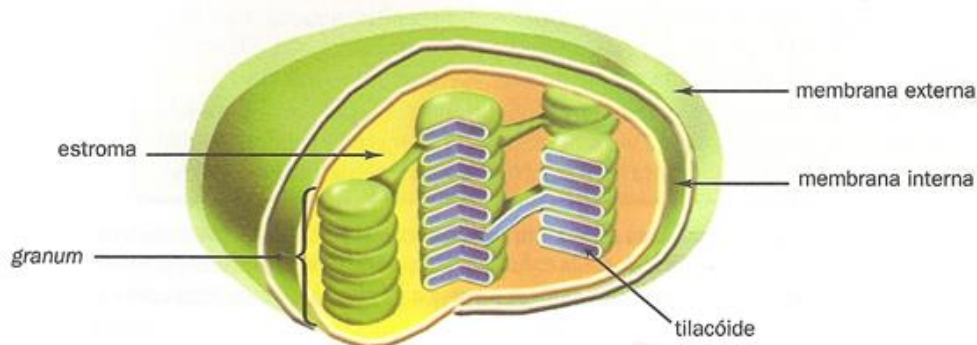


Figura 30 - Desenho esquemático de cloroplasto visto em corte
FONTE: (LAURENCE, 2005, p. 142)

Segundo Laurence (2005), a fotossíntese pode ser dividida na fase fotoquímica e química (Figura 31).

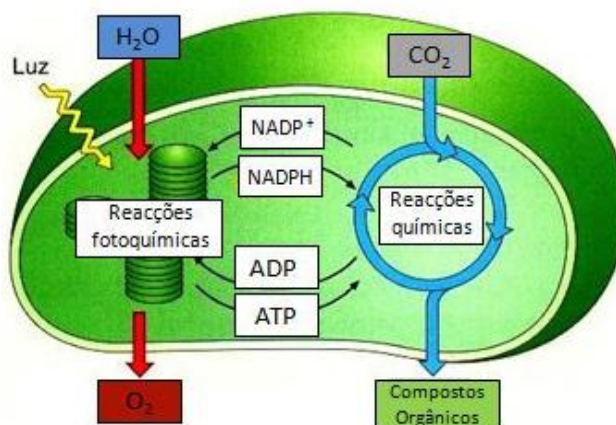


Figura 31 - Etapas da fotossíntese
FONTE: Disponível no site Biologia e Geologia.¹⁵

2.4.3.1 Fase Fotoquímica

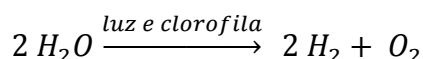
A fase fotoquímica é a fase dependente da luz. Nela, há a participação das clorofilas e ocorrem a fotólise da água e a fotofosforilação cíclica e acíclica. Todo o

¹⁵ Disponível em <<http://10anocvg.blogspot.com.br/2013/06/fotossintese.html>>. Acesso em: 10 ago. 2013.

processo ocorre nas tilacóides (sistema membranosos interno do cloroplasto em forma de vesículas achatadas, que contém as proteínas necessárias para a fotossíntese).

Fotólise da água

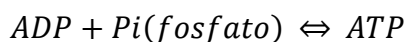
A fotólise é o processo de decomposição de uma substância mais complexa em duas substâncias mais simples por ação da luz. A primeira etapa da fotossíntese consiste em uma reação química de análise (ou decomposição), realizada somente na presença de luz e de clorofila, em que a água é decomposta em hidrogênio e oxigênio:



Nessa primeira etapa, há a liberação do oxigênio e o hidrogênio é transferido para moléculas de NADP para ser utilizado na fase escura da fotossíntese. Deste processo, há a formação de NADPH₂.

Fotofosforilação

O processo de fotofosforilação significa adição de fosfato em presença de luz (*foto*). A substância solúvel existente no citoplasma das células, ADP (adenosina difosfato ou difosfato de adenosina), é adicionada ao fosfato, formando o ATP (adenosina trifosfato ou trifosfato de adenosina), que também será utilizada na próxima fase da fotossíntese. A clorofila participa deste processo.



As moléculas de ATP armazenam grande parte da energia liberada na oxidação de moléculas orgânicas nas células, e atuam como “moedas energéticas” a serem gastas em processos celulares.

Em termos de estrutura, o ATP é um nucleotídeo, ou seja, composto formado por uma base nitrogenada, uma pentose e um grupo fosfato. Neste caso, a sua base nitrogenada é a adenina, a molécula de açúcar é a ribose e o grupo fosfato é composto por três moléculas de ácido fosfórico (H_3PO_4).

Grande parte da energia liberada nas reações exotérmicas ficam armazenadas nas ligações-fosfato. No ATP, esta energia fica especialmente armazenada na última

ligação fosfato, sendo liberada quando ela é quebrada. Quando ocorre esta liberação de energia, o ATP transforma-se em ADP. Analogamente falando, esse processo é como o funcionamento da bateria de um telefone celular. Carregar a bateria na tomada significaria transformar o ADP em ATP. Descarregar a bateria, gastando sua energia, corresponderia à passagem do ATP para o ADP. (QIEDUCACAO, 2013)

2.4.3.2 Fase Química

A fase química ocorre na matriz ou estroma dos cloroplastos e não há necessidade direta da luz ou clorofila, mas precisa da energia armazenada no ATP produzido na fotofosforilação, dos hidrogênios provenientes da fotólise de água, além do CO_2 do ar atmosférico e aquele produzido pela sua própria respiração.

Nesta fase, as moléculas de CO_2 e hidrogênio ligadas aos seus transportadores ($NADPH_2$) participam de um ciclo chamado de Calvin-Benson, e dão origem às moléculas de carboidrato $(CH_2O)_n$ e água. A molécula sofre polimerização (processo que consiste na união de moléculas de um dado composto para formar um novo composto) imediata, e dá início à produção de açúcares simples, principalmente a glicólise ($C_6H_{12}O_6$).

2.4.4 A física na fotossíntese

Um triunfo da física no início do século XX foi a descoberta que a luz possui características tanto de partículas quanto de ondas. Uma onda é caracterizada por um comprimento de onda, representado pela letra grega lambda (λ), que é a distância entre dois picos sucessivos.

A frequência, representada pela letra grega nu (η), é o número de picos que passam por um observador em um dado tempo. Uma equação simples relaciona o comprimento, a frequência e a velocidade de qualquer onda: $\eta\lambda = c$ onde, η é a frequência, λ é o comprimento de onda e c é a velocidade da luz

A luz também é uma partícula, a qual denominamos fóton. Cada fóton contém uma quantidade de energia que é chamada de quantum (plural: quanta). O conteúdo de energia de luz não é contínuo, mas emitido nesses pequenos “pacotes”, os *quanta*.

A energia (E) de um fóton depende da frequência, de acordo com a relação conhecida como a lei de Planck:

$$E = h\eta$$

Sendo,

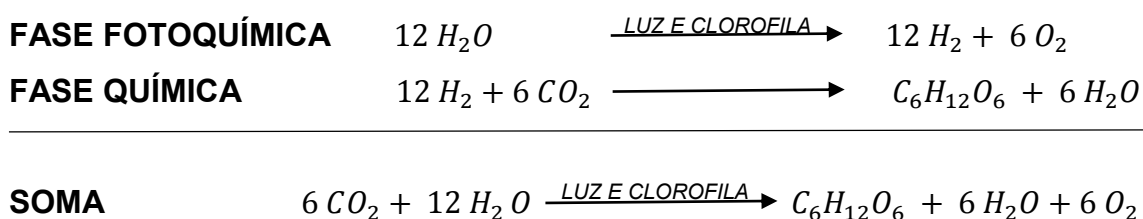
h a Constante de Planck; ($h = 6,63 \times 10^{-34} \text{ J.s}$)

η a frequência;

E a energia do fóton.

2.4.5 A química na fotossíntese

A representação geral do processo fotossintético pode ser obtida pela soma das equações simplificadas das duas fases da fotossíntese:



Na representação demonstrada acima, o ATP e ADP não foram considerados, pois o que é produzido na fase clara é consumido na fase escura.

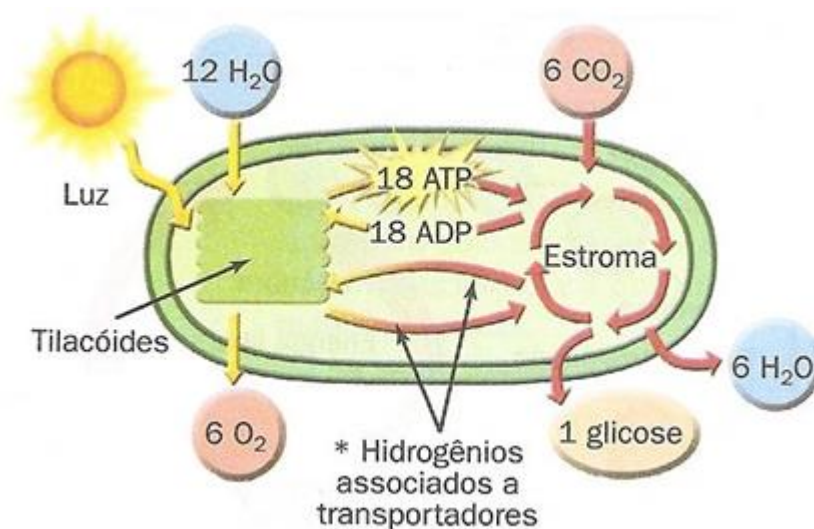


Figura 32 - Esquema de fotossíntese nos cloroplastos.
 FONTE: (LAURENCE, 2005, p. 144)

3. SOLUÇÃO PROPOSTA

Tendo em vista o Trabalho de Conclusão de Curso (TCC) foi proposto o desenvolvimento de um aplicativo para *tablets* Android, onde foram utilizadas técnicas para construção de um objeto de aprendizagem focado ao ensino de conceitos científicos, mais especificamente o processo de fotossíntese, para alunos do Ensino Médio. Para isso foram utilizados textos informativos com linguagem didática e de fácil compreensão, imagens ilustrativas e animações das mesmas. Com o objetivo de proporcionar uma maior interação com o usuário, utilizou-se o sensor de luminosidade para criação de uma atividade referente ao processo da fotossíntese.

O projeto de construção do OA foi dividido em fases, conforme proposto pelo modelo ADDIE (Figura 33). Abaixo estas serão detalhadamente descritas.



Figura 33 - Fases de desenvolvimento do projeto de TCC.
 FONTE: Elaborada pela autora.

3.1 Fase de Análise

Na primeira etapa do projeto realizou-se um estudo dos sensores de hardware presentes em dispositivos com o sistema operacional Android, com o objetivo de compreender o que os dados vindos dos sensores representam e como são

produzidos. As documentações consultadas em Milette e Stroud (2012) e Android Developers (2013) foram bastante completas e proporcionaram um rápido aprendizado sobre a utilização dos mesmos.

Posteriormente, realizou-se um estudo sobre vários conceitos dos Objetos de Aprendizagem, tais como definições, aspectos pedagógicos, modelo de desenvolvimento, exemplos e pesquisas já realizadas sobre o assunto.

A partir dos estudos feitos, foi possível analisar qual conceito científico do ensino de Ciências Naturais poderia ser explorado. Optou-se por trabalhar com a fotossíntese - processo de síntese que exige luz para que se realize (foto = luz). Para explicar o seu funcionamento e oferecer um OA interativo, capaz de atrair a atenção do aluno, utilizou-se o sensor de luminosidade, presente na maioria dos dispositivos Android.

3.2 Fase de Projeto

Em posse das informações obtidas na etapa anterior, foi desenvolvido um estudo sobre quais conceitos científicos referentes à fotossíntese deveriam ser abordados neste OA. Entre eles, constam:

- Definição e importância: apresentação dos conceitos científicos relacionados ao tema e sua importância para o meio ambiente;
- Histórico: contextualização da história da fotossíntese, apresentando os cientistas envolvidos na descoberta deste processo;
- Aparelho fotossintético: detalhamento da estrutura celular responsável para a realização da fotossíntese;
- Processo da fotossíntese: descrição das etapas envolvidas no processo. O sensor de luminosidade é empregado nesta etapa para criar as animações referentes ao que acontece com a planta na presença ou ausência de luz.
- A física na fotossíntese: explicação sobre o processo físico envolvido na fotossíntese;
- A química na fotossíntese: explicação sobre o processo químico envolvido na fotossíntese.

Após estudo e definição dos tópicos abordados, organizou-se um diagrama com as telas do aplicativo (Figura 34).

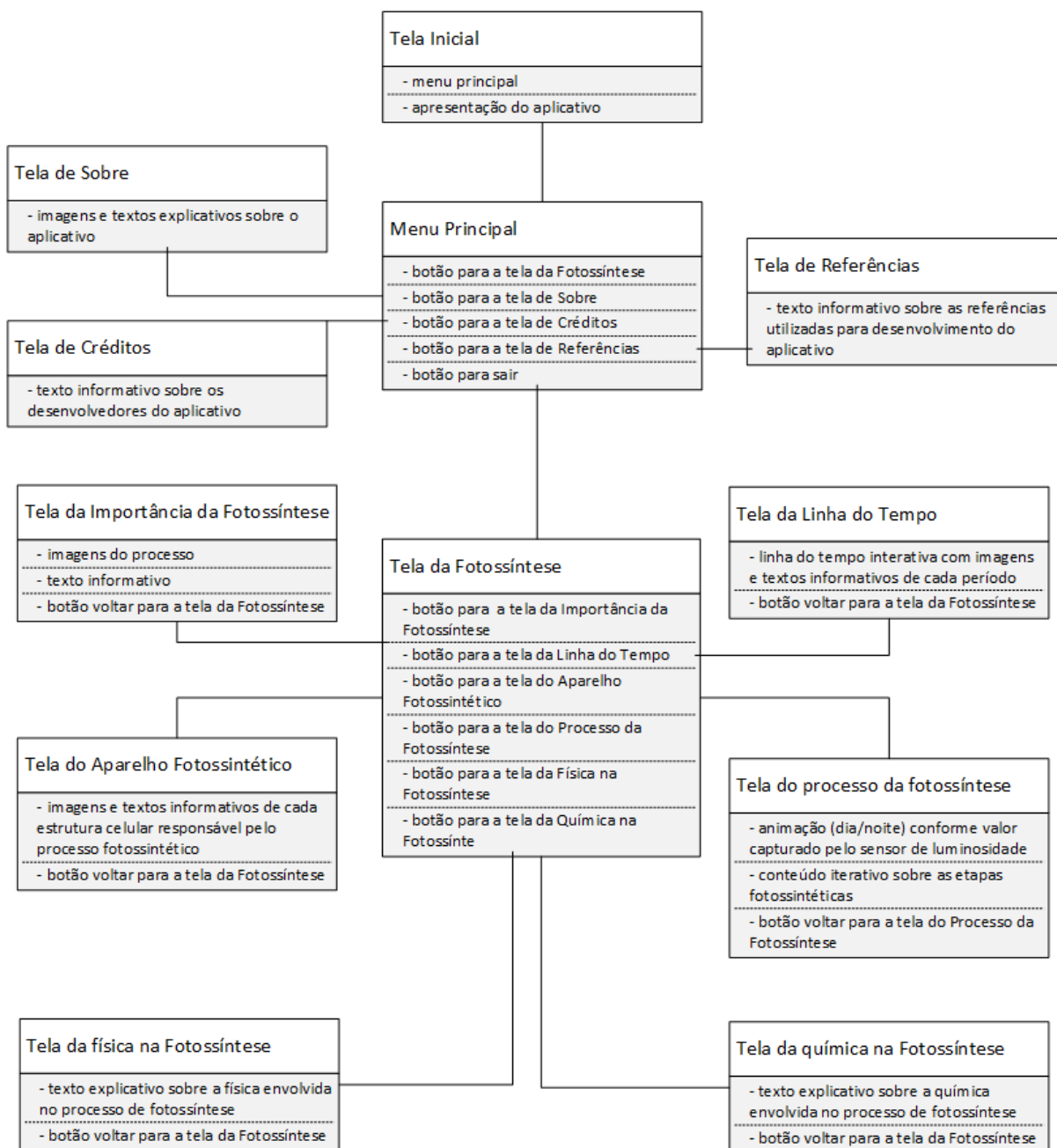


Figura 34 - Diagrama das telas do aplicativo.

FONTE: Elaborada pela autora, com o auxílio da designer do projeto.

Conforme visto na Figura 34, o objeto apresenta 11 (onze) telas, todas focadas na apresentação prática dos conceitos envolvidos no processo da fotossíntese, através de textos informativos, imagens e animações das mesmas.

Com o conteúdo teórico e alguns parâmetros de programação delineados, foram feitos os primeiros rascunhos das telas pela aluna de Designer, com a composição dos textos e suas funcionalidades (Figura 35).



Figura 35 - Imagens das telas provisórias do projeto – Tela da Linha do tempo e Aparelho fotossintético. FONTE: Imagens elaboradas pela designer do projeto.

Após a análise e pesquisa dos componentes Android e refinamento da interação do usuário, as telas foram se aperfeiçoando (Figura 36).

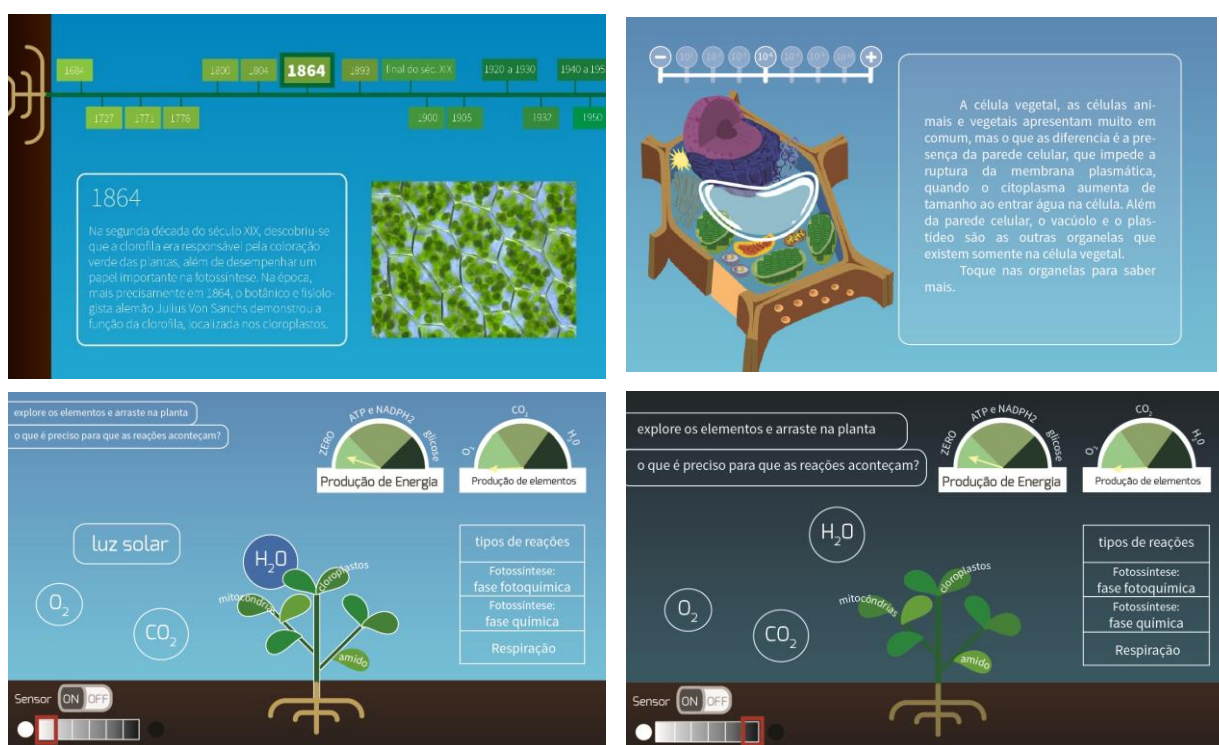


Figura 36 – Representação de algumas telas do protótipo do aplicativo - Tela da Linha do tempo, Aparelho fotossintético e Telas do Processo da fotossíntese (à esquerda, etapa fotoquímica e à direita, etapa química). FONTE: Imagens elaboradas pela designer do projeto.

3.3 Fase de Desenvolvimento

Ferramentas e componentes utilizados

Para iniciar o desenvolvimento de aplicativos em Android é necessário configurar um ambiente contendo a última versão do Java Development Kit (JDK), juntamente com o Android SDK e uma IDE para codificação. (GONÇALVES, 2011).

Para o projeto desenvolvido, foram utilizadas as seguintes ferramentas:

- JDK 7;
- Android SDK Tools;
- IDE Eclipse na sua versão 4.2 (codinome Juno) + ADT *plug-in* (utilizado para integrar o Android SDK ao Eclipse);
- Plug-in ObjectAid UML Explorer para o Eclipse, utilizado para construção do diagrama de classes do projeto.¹⁶

Os detalhes referentes à instalação e configuração do ambiente de desenvolvimento em Android podem ser encontradas na documentação oficial do sistema, presente em Android Developers (2013).

Para a implementação foi utilizada a linguagem Java, indicada para desenvolvimento de aplicações na plataforma Android. Como o objetivo do projeto é desenvolver uma aplicação para *tablets*, a versão mínima do sistema é o Android 3.0 (Honeycomb), pois foi a partir dela que a interface do usuário foi otimizada para tais dispositivos.

Abaixo são apresentados os principais componentes Android, consultados na documentação oficial do sistema, que foram utilizados para compor o *layout*:

Activity: uma *activity* é o componente principal de um aplicativo Android. Cada *activity* possui uma tela onde é desenhada a interface do usuário e é através dela que o usuário pode interagir com a aplicação para fazer algo, tal como realizar ligação, tirar uma foto, enviar e-mail, visualizar um mapa.

¹⁶ As instruções para uso e instalação do *plug-in* podem ser encontradas no site: <http://www.objectaid.com/>.

Todo aplicativo Android inicia por uma *activity*. A classe *Activity* define as seguintes séries de eventos que governam o ciclo de vida de uma *activity*:

- `onCreate()` – chamado quando a *activity* é criada;
- `onRestart()` – chamado sempre quando uma *activity* é interrompida antes de executar novamente;
- `onStart()` – chamado quando a *activity* está se tornando visível para o usuário;
- `onPause()` – chamado quando o sistema está prestes a retornar a uma *activity* anterior;
- `onStop()` – chamado quando uma *activity* já não está mais visível para o usuário;
- `onResume()` – chamado quando a *activity* recomeça a interagir com o usuário;
- `onDestroy()` – chamado antes da *activity* ser destruída pelo sistema para liberar espaço em memória.

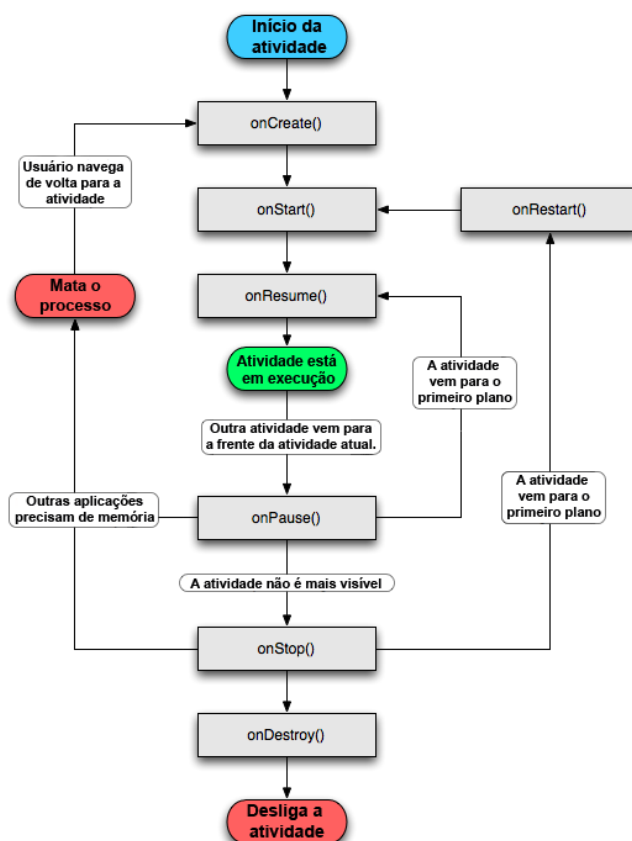


Figura 37 - Ciclo de vida de uma *activity*

FONTE: Adaptação da imagem presente no site Vitor Meriat¹⁷

¹⁷ Disponível em <<http://vitormeriat.files.wordpress.com/2011/11/ciclo-de-vida.png>>. Acesso em 4 ago. 2013.

Fragment: foram introduzidas na versão Android 3.0, para tornar a interface do usuário mais dinâmica e flexível em telas maiores, tais como *tablets*.

Um *fragment* representa o comportamento de uma parte da interface do usuário em uma *activity*. Pode ser visto como uma seção modular de uma *activity*, que possui seu próprio ciclo de vida, recebe seus próprios eventos de entrada, e pode ser adicionado ou removido enquanto a *activity* está em execução, podendo também, ser reutilizado em várias delas.

Os *fragments* devem ser projetados de maneira modular e reutilizáveis. Isso possibilita o desenvolvimento de aplicativos que devem suportar múltiplos *layouts*, tais como *tablets* e *smartphones*, pois permite posicionamento dos objetos conforme o tamanho da tela do dispositivo do usuário. A Figura 38 apresenta um exemplo da utilização desse componente. No caso da versão para *tablets*, a *Activity A* pode comportar dois *fragments* para compor a interface do usuário e preencher toda a tela. No entanto, para *smartphones* não há espaço suficiente para os dois *fragments*, sendo necessário utilizar duas *activities* para comportá-los.

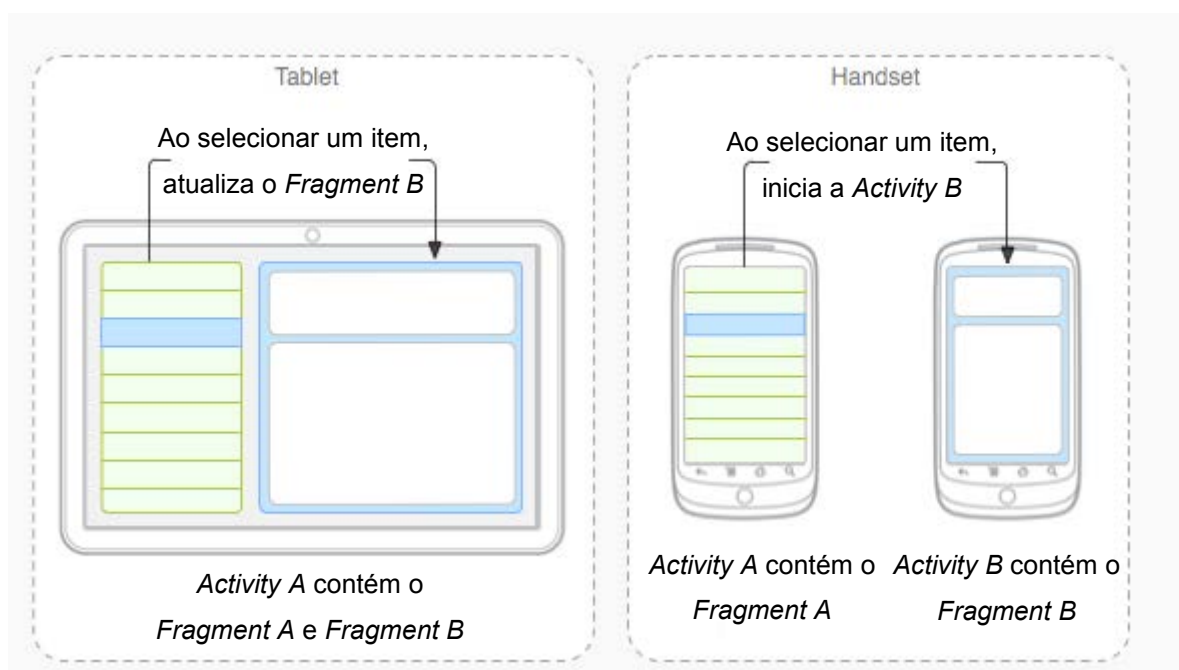


Figura 38 - Exemplo da utilização dos *fragments* na construção de *layouts* para *tablets* e *smartphones*
 FONTE: Adaptação da imagem disponível na documentação oficial do Android.¹⁸

¹⁸ Disponível em <<http://developer.android.com/guide/components/fragments.html>>. Acesso em 4 ago. 2013.

Navigation Drawer: é um componente utilizado para facilitar a navegação multinível em uma aplicação Android. O usuário pode arrastar o painel de menu da lateral da tela e selecionar um dos itens. É recomendado quando há quatro ou mais itens de menu. A Figura 39 mostra um exemplo da utilização do componente.



Figura 39 - Exemplo de aplicativo utilizando o Navigation Drawer
 FONTE: Disponível no site da documentação oficial do Android.¹⁹

ViewPager: é um componente que permite ao desenvolvedor apresentar uma lista horizontal de “páginas”, sendo que cada página representa um *fragment*. Normalmente é utilizado em apresentações de slides para realizar a transição de uma tela para a outra, através da interação do usuário (deslizando o dedo sobre a tela) ou programaticamente. Utiliza um *adapter*, *PagerAdapter*, para preencher os dados relativos a cada uma das páginas.

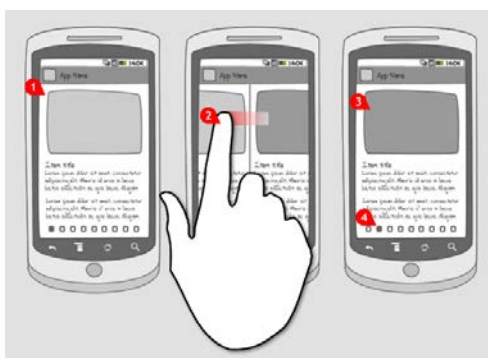


Figura 40 - Exemplo de utilização do ViewPager.
 FONTE: Disponível no site Android Patterns.²⁰

¹⁹ Disponível em <
http://developer.android.com/design/media/navigation_drawer_overview.png>. Acesso em 20 ago. 2013.

²⁰ Disponível em <
<http://www.androidpatterns.com/wp-content/uploads/Horizontalscrolling2.png>>. Acesso em 20 ago. 2013.

ListView: componente utilizado para exibir uma lista de itens escroláveis. Os itens são automaticamente inseridos na lista utilizando um *adapter*, responsáveis por preencher item por item desta lista.



Figura 41 - Exemplo de utilização do ListView.
 FONTE: Disponível no site Vogella.²¹

DragAndDrop: o componente, disponível para desenvolvedores a partir da API 11 (Android 3.0 HoneyComb) é uma forma interessante de interação com o usuário. Com sua utilização, o usuário é capaz de arrastar um componente com o dedo para determinada área. Um exemplo é encontrado na exclusão de um aplicativo em um sistema Android. Para excluí-lo, basta manter o dedo pressionado sobre o ícone do aplicativo alvo e arrastar até a lixeira.

Interface

O aplicativo é composto por 11 (onze) telas. A primeira delas é a tela inicial, que contém a apresentação do aplicativo e um menu de opções. Para a construção deste, foi utilizado o componente *Navigation Drawer*, de forma a ganhar espaço disponível na tela. Neste padrão, o menu fica escondido e ao arrastar o dedo do lado esquerdo para o direito na tela ou ao clicar no ícone que fica no canto superior esquerdo, próximo ao ícone do aplicativo, se abre a lista de itens disponíveis. O movimento contrário (do lado direito para o esquerdo) esconde o menu.

A lista de opções contém cinco itens principais, que servirão para:

- Iniciar as atividades referentes ao conceito físico explorado;

²¹ Disponível em http://www.vogella.com/tutorials/AndroidListView/images/xlistview_scheme10.png.pagespeed.ic.oP5IBQ-2EL.png. Acesso em: 20 ago. 2013.

- Acessar a tela que contém as informações necessárias para a utilização do aplicativo e explicações detalhadas sobre o sensor de luminosidade;
- Acessar a tela que contém as informações referentes aos desenvolvedores do projeto;
- Acessar a tela contendo as referências bibliográficas consultadas para o desenvolvimento do objeto;
- Encerrar a aplicação;

Ao clicar sobre o item 'Fotossíntese', é apresentada uma tela com vários botões para acesso aos conceitos explorados sobre a fotossíntese:

- *Importância da fotossíntese*: nela, o usuário poderá compreender, por meio de uma animação referente ao valor captado pelo sensor de luminosidade (dia ou noite) e textos informativos, qual é a importância do processo fotossintético.

- *Linha do tempo*: através de textos informativos e imagens, apresenta as datas referentes a cada descoberta científica sobre o processo da fotossíntese e os cientistas responsáveis por elas. Para percorrer a linha do tempo, o usuário poderá clicar na data presente logo no topo da tela ou realizar o movimento de mudança de página nos textos que descrevem as descobertas, localizados abaixo das datas.

- *Aparelho fotossintético*: nessa tela é apresentada a imagem de uma planta e alguns índices sobre as suas estruturas celulares. Ao clicar em algum deles, é apresentado textos e imagens referentes àquela estrutura e qual é a sua participação no processo da fotossíntese. Na última tela, há uma animação referente às fases processo de fotossíntese.

- *Processo da fotossíntese*: nela ocorrem as atividades referentes ao processo da fotossíntese. Contém um botão para ligar/desligar o sensor de luminosidade, três botões que representam este processo (Fase fotoquímica, Fase química e Respiração), elementos necessários para a realização das etapas (luz solar, H_2O , CO_2 e O_2), um indicador de produção de energia da planta que possui três fases (1 – zero, 2 – ATP e NADPH₂, 3 – glicose) e um indicador de produção de elementos (H_2O , CO_2 e O_2).

Conforme o valor obtido pelo sensor, o cenário da tela é alterado (dia/noite) e para cada fase apresentada no indicador de produção de energia, o usuário deve arrastar para a planta o elemento necessário de sua etapa correspondente (Quadro 5).

Quadro 5 - Etapas do processo da fotossíntese demonstradas no aplicativo.

Estado do sensor	Fase do dia	Fase da produção de energia	Elementos necessários	Fase da fotossíntese	Resultado
Ligado	Dia	Fase 1 (Zero)	H_2O e luz solar	Fase Fotoquímica	Liberação de oxigênio (O_2) e produção de energia fase 2
	Noite		-	-	-
	Dia	Fase 2 (ATP e $NADPH_2$)	CO_2 , ATP e $NADPH_2$	Fase Química	Produção de energia fase 3
	Noite		-	-	-
	Dia ou Noite	Fase 3 (glicose)	O_2 e glicose	Respiração	Planta faz respiração
Desligado	Noite	Fase 1 (Zero)	-	-	-
		Fase 2 (ATP e $NADPH_2$)	-	-	-
		Fase 3 (glicose)	O_2 e glicose	Respiração	Planta faz respiração

FONTE: Elaborada pelas desenvolvedora, com auxílio da designer do projeto.

A primeira etapa da fotossíntese, por exemplo, somente se realiza em presença de luz. Assim, se a produção de energia estiver na fase 1 (Zero), o valor obtido pelo sensor for equivalente ao período noturno, o cenário ficará escuro e caso o usuário arraste qualquer um dos elementos presentes na tela, nada acontecerá. Caso contrário, o cenário ficará claro e quando o usuário arrastar a água (H_2O) e a luz solar para a planta, o botão da Fase Fotoquímica acende, o oxigênio (O_2) é liberado e ocorre a produção de energia fase 2 (ATP e $NADPH_2$).

- *Física na fotossíntese*: apresentação dos conceitos físicos realizados na fotossíntese, por meio de textos informativos e equações.

- *Química na fotossíntese*: apresentação dos conceitos químicos realizados na fotossíntese, por meio de textos informativos e equações.

Programação

Todo projeto Android contém um arquivo XML chamado '*string.xml*', localizado no diretório raiz do projeto (*res/values/string.xml*). Nele, são armazenados os textos exibidos no aplicativo, como por exemplo, nome do aplicativo, nome dos botões e títulos das *activities*. Para facilitar a manutenção do aplicativo, todos os textos ficarão armazenados neste arquivo.

Como exemplo de sua utilização, pode-se citar os itens do menu de opções do aplicativo. Como dito anteriormente, todos encontram-se armazenados no arquivo '*string.xml*' (Figura 42).

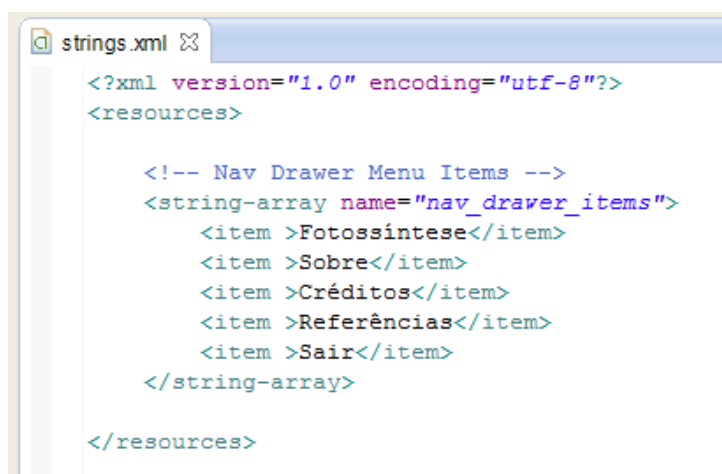


Figura 42 - Representação do uso do arquivo *string.xml*.

FONTE: Imagem elaborada pela autora com base no arquivo *string.xml* do projeto.

A leitura destes itens armazenados no arquivo *string.xml* é feita da seguinte maneira:

```
String[] navMenuTitles = getResources().getStringArray(R.array.nav_drawer_items);
```

Figura 43 - Trecho de código, extraído do projeto, responsável por obter os dados do tipo String do arquivo *string.xml*.

FONTE: Elaborada pela autora com base no trecho de código contido na classe *MainActivity*.

A estrutura do projeto pode ser analisada segundo o diagrama de classes, que pode ser visto no “APÊNDICE A” deste projeto. O aplicativo é composto por dezoito classes. A classe *MainActivity* herda da classe *Activity* e apresenta a *activity* principal do projeto. Todas as outras classes com a terminação *Fragment*, herdam da classe *Fragment* e apresentam as telas de funcionalidades do mesmo.

A classe *NavDrawerListAdapter* herda da classe *BaseAdapter* e trabalha como um adaptador que recebe um *array* de objetos da classe *NavDrawerItem* e preenche a lista do menu de opções.

A classe *LinhaTempoAdapter* herda da classe *PagerAdapter* e trabalha como um adaptador que recebe um *array* de objetos do tipo *String* e preenche as informações referentes a cada contribuição científica para o avanço do conhecimento sobre o processo da fotossíntese.

A classe *ProcessoFotossinteseFragment* implementa a interface *SensorEventListener*, usada para receber notificações do *SensorManager* quando ocorrem mudanças no valor dos sensores registrados (ANDROID DEVELOPERS, 2013).

SensorManager é a classe responsável por gerenciar o acesso da aplicação aos sensores internos dos dispositivos Android. Para acessá-la é necessário manter uma instância de classe, que pode ser adquirida ao chamar o método *getActivity().getSystemService(Context.SENSOR_SERVICE)* e a partir dela, obter acesso ao sensor de luminosidade. A Figura 44 mostra o método *onCreateView()* do *fragment* contendo as instâncias para a *SensorManager* e o sensor de luminosidade.

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    View rootView = inflater.inflate(
        R.layout.fragment_processo_fotossintese, container, false);

    mSensorManager = (SensorManager) getActivity().getSystemService(
        Context.SENSOR_SERVICE);

    mSensorLuminosidade = mSensorManager
        .getDefaultSensor(Sensor.TYPE_LIGHT);

    return rootView;
}
```

Figura 44 - Método *onCreateView()* da classe *ProcessoFotossinteseFragment*
 FONTE: Elaborado pela autora com base no trecho de código extraído do projeto.

Antes de registrar um *SensorEventListener*, responsável por receber as notificações do *SensorManager*, é preciso verificar se o sensor está presente no dispositivo. Caso esteja, é registrado um *SensorEventListener* para o sensor de luminosidade. Caso contrário, o sistema apresenta uma mensagem de erro para o usuário informando que o sensor não encontra-se disponível e o *fragment* é encerrado. A Figura 45 apresenta o método *onResume()*, onde é feita essa verificação.

```
@Override
public void onResume() {
    super.onResume();

    if (sensorExiste()) {
        mSensorManager.registerListener(this,
            mSensorLuminosidade,
            SensorManager.SENSOR_DELAY_NORMAL);
    }
    else {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("Erro");
        builder.setMessage(
            "Seu dispositivo não possui o sensor de luminosidade")
            .setCancelable(false)
            .setPositiveButton("Ok",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,
                        int id) {
                        getActivity()
                            .getFragmentManager()
                            .beginTransaction()
                            .remove(ProcessoFotossinteseFragment.this)
                            .commit();
                    }
                });
        AlertDialog alert = builder.create();
        alert.show();
    }
};

private boolean sensorExiste() {
    List<Sensor> sensor = mSensorManager.getSensorList(Sensor.TYPE_LIGHT);
    return sensor.size() > 0;
}
```

Figura 45 - Trecho do código extraído da classe *ProcessoFotossinteseFragment*.
FONTE: Elaborado pela autora.

Com o *listener* registrado, o *fragment* deve implementar o método *onSensorChanged()* (Figura 46). Esse método é invocado sempre que ocorrem mudanças nos valores dos sensores registrados. É nele que o valor do sensor de

luminosidade será obtido a cada 500 milissegundos e implementadas as ações referentes ao que ocorre no processo de fotossíntese na ausência ou presença de luz.

```
@Override
public void onSensorChanged(SensorEvent event) {

    if (event.sensor.getType() == Sensor.TYPE_LIGHT) {

        long tempoAtual = System.currentTimeMillis();

        if (tempoAtual - ultimaAtualizacao > 500) {

            ultimaAtualizacao = tempoAtual;
            float valorSensorLuminosidade = event.values[0];
            atualizaProcessoFotossintese(valorSensorLuminosidade);
        }
    }
}
```

Figura 46 - Implementação do método `onSensorChanged()` na classe *ProcessoFotossinteseFragment*.
FONTE: Elaborada pela autora com base no trecho de código extraído do projeto.

As classes *ChoiceDragListener*, *ChoiceOnLongClickListener* e *ChoiceTouchListener* são utilizadas na classe *ProcessoFotossinteseFragment* para implementar os *listeners* referentes a ação de arrastar um componente para determinada posição da tela, efetuar um clique longo (manter o dedo pressionado sobre um item) e evento de clique em algum ícone, respectivamente.

3.4 Fases de Implementação e Validação

Para a validação do protótipo foram realizados testes exploratórios no aplicativo, com o propósito de explorar todas as funcionalidades e encontrar o máximo de falhas. Todas elas foram testadas diversas vezes sendo que em todas, foram obtidos os resultados esperados.

Após análise das funcionalidades, a designer gráfica do projeto avaliou o aplicativo quanto a interface do aplicativo. Segundo ela, o resultado obtido com a arte final foi satisfatório e adequado às diretrizes do desenvolvimento Android.

A aplicação foi validada no *tablet* Samsung Tab 2 P3110, com tela de sete polegadas, 8 GB de memória, processador Dual Core 1.0 GHz e sistema operacional Android 4.1.2 (Jelly Bean).



Figura 47 - *Tablet* utilizado para testes do aplicativo
FONTE: Disponível no site da Samsung.²²

²² Disponível em <<http://www.samsung.com/br/consumer/cellular-phone/cellular-phone-tablets/tablet/GT-P3100ZWMZTA>>. Acesso em: 10 dez. 2013.

4. CONCLUSÃO

Este trabalho apresentou a construção de um objeto de aprendizagem em forma de aplicativo para *tablets* Android, que explore os sensores internos existentes nestes dispositivos, com o objetivo de facilitar o ensino-aprendizagem do processo da fotossíntese para os estudantes do Ensino Médio.

O projeto consistiu em uma série de estudos sobre o funcionamento dos sensores presentes nos dispositivos da plataforma Android. Através desta compreensão e análise dos dados obtidos pelos mesmos, foi definida uma aplicação que fizesse uso do sensor de luminosidade para explicar os conceitos envolvidos no processo da fotossíntese, a fim de proporcionar ao aluno maneiras de observar e entender como o ambiente interfere neste processo.

Os passos seguintes consistiram em mais estudos sobre os assuntos necessários para a criação do mesmo, tais como definições, aplicações e metodologia de desenvolvimento de objetos de aprendizagem e compreensão da fotossíntese.

O aplicativo desenvolvido consta de uma interface de usuário atraente e fácil de usar, com textos informativos, escritos em linguagem clara e didática, imagens e animações, além de uma atividade interativa que emprega a utilização do sensor de luminosidade para demonstrar o processo da fotossíntese.

O objeto final comprovou cumprir seus objetivos conforme as validações realizadas, alcançando os objetivos propostos para o projeto. Com o uso desta ferramenta espera-se contribuir com a produção de objetos de aprendizagem, fazendo com que o conceito do processo de fotossíntese abordado na disciplina de Ciências para o Ensino Médio seja melhor compreendido pelos alunos e sirva, também, como material de apoio em sala de aula para professores.

Como trabalho futuro pretende-se avaliar os verdadeiros resultados da aplicação do uso deste aplicativo em salas de aula, através de pesquisas, de forma a analisar a eficiência e eficácia do objeto desenvolvido e opinião dos alunos e professores a respeito do material.

4.1 Contribuições e trabalhos futuros

Neste trabalho foram exploradas diversas características relativas aos sensores internos dos dispositivos Android, procurando contribuir com um conteúdo de leitura aos interessados em explorar tais recursos, através da apresentação de um material bastante completo e com informações que facilitam o desenvolvimento de uma aplicação que faça uso destes.

Além disso, pode ser utilizado como base para a criação de um projeto maior contendo vários outros objetos de aprendizagem sobre assuntos diversos, como Química, Física e Matemática, que explorem outros sensores internos disponíveis em dispositivos como *tablets* e *smartphones*.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, R. **Distribuição do Android em Dezembro, Jelly Bean com 54,5% e KitKat com apenas 1.1%.** 2013. Disponível em <<http://mobilexpert.com.br/apps/outros/materias/6693/distribuicao-do-android-em-dezembro-jelly-bean-com-545-e-kitkat-com-apenas-11>>. Acesso em: 15 dez. 2013.

ANDRADE, A. P. R. **O Uso das Tecnologias na Educação: Computador e Internet.** 2011. Disponível em <<http://www.fe.unb.br/catedraunescoead/areas/menu/publicacoes/monografias-sobre-tics-na-educacao/o-uso-das-tecnologias-na-educacao-computador-e-internet>>. Acesso em: 20 mar. 2013.

ANDROID DESIGN. **UI Overview.** 2013. Disponível em <<https://developer.android.com/design/get-started/ui-overview.html>>. Acesso em: 20 nov. 2013.

ANDROID DEVELOPERS. **Environment Sensors.** 2013a. Disponível em: <http://developer.android.com/guide/topics/sensors/sensors_environment.html>. Acesso em: 10 abr. 2013.

ANDROID DEVELOPERS. **Motion Sensors.** 2013b. Disponível em: <http://developer.android.com/guide/topics/sensors/sensors_motion.html>. Acesso em: 10 abr. 2013.

ANDROID DEVELOPERS. **Position Sensors.** 2013c. Disponível em: <http://developer.android.com/guide/topics/sensors/sensors_position.html>. Acesso em: 10 abr. 2013.

ANDROID DEVELOPERS. **Sensors Overview.** 2014d. Disponível em: <http://developer.android.com/guide/topics/sensors/sensors_overview.html>. Acesso em: 10 abr. 2013.

ARANTES, A. R.; MIRANDA, M. S.; STUDART, N. **Objetos de aprendizagem no ensino de física: usando simulações do PhET.** Física na Escola, v.11, n.1, 2010. Disponível em <<http://www.sbfisica.org.br/fne/Vol11/Num1/a08.pdf>>. Acesso em: 31 maio 2013.

ASSIS, K. K.; CZELUSNIAK, S. M.; ROEHRIG, S. A. G. **A articulação entre o ensino de Ciências e as TICs: Desafios e possibilidades para a formação continuada.** 2011. Disponível em: <http://educere.bruc.com.br/CD2011/pdf/5209_2477.pdf>. Acesso em: 25 nov. 2013.

BARDY, Livia Raposo; HAYASHI, Maria Cristina Piumbato Innocentini; SCHLUNZEN, Elisa Tomoe Moriya; SEABRA JUNIOR, Manoel Osmar. **Objetos de Aprendizagem como recurso pedagógico em contextos inclusivos: subsídios para a formação de professores a distância.** *Rev. bras. educ. espec.* [online]. 2013, vol.19, n.2, p. 273-288.

BETTIO, R. W. **Avaliações Gráficas e Dinâmicas Aplicadas a Ambientes Virtuais de Aprendizagem.** 2003. Disponível em <<https://repositorio.ufsc.br/bitstream/handle/123456789/84571/190727.pdf?sequence=1>>. Acesso em: 30 maio 2013.

BORGES, F.; NAVARRO, M. **Aplicação colaborativa de objetos de aprendizagem, a partir de uma proposta de planejamento pedagógico integrado.** Belo Horizonte. 2005. Disponível em: <http://portal.ibta.com.br/cursos/ibtanews/news-01-06/downloads/objetos_pgl.doc>. Acessado em: 30 maio 2013.

BRASIL. Ministério da Educação. Secretaria da Educação Fundamental. **Parâmetros Curriculares Nacionais: 1ª a 4ª série - Introdução.** Brasília: MEC/SEF, 2000.

BUDZINSKI, J. A. S. **Acelerômetros.** 1999. Disponível em: <<http://www.eletrica.ufpr.br/edu/Sensores/1999/joao/funcionamento.htm>>. Acesso em: 14 abr. 2013.

CARDOSO, V. M; CORDENONZI, W; GASPARY, L. M; MARQUEZ, G. R. **OAMVet – Desenvolvimento de um Objeto de Aprendizagem para o estudo de vetores.** 2013. Disponível em <<http://www.conferencias.ulbra.br/index.php/ciem/vi/paper/viewFile/820/623>>. Acesso em: 31 maio 2013.

CASTRO, J. A. **Objetos de aprendizagem e sua utilização no ensino de matemática.** Disponível em <http://paginapessoal.utfpr.edu.br/kalinke/grupos-de-pesquisa/pde/pdf/objetos_de_aprendizagem_e_EM.pdf>. Acesso em: 31 maio 2013.

CORTONI, S. Z. **Jovens são alvos da tecnologia.** 2006. Disponível em <<http://www.administradores.com.br/noticias/administracao-e-negocios/jovens-sao-alvos-da-tecnologia/7307/>>. Acesso em: 20 mar. 2013.

CRUZ, F. C. **Erros e incertezas, e dicas sobre relatórios.** Disponível em: <<http://www.ifi.unicamp.br/leb/f32909/erros%20ou%20incertezas%20e%20dicas%20sobre%20relatorios.pdf>>. Acesso em: 23 abr. 2013.

DIAS, C. C. L.; KEMCZINSKI, A.; LUCENA, S. S.; FERLIN, J.; HOUNSELL, M. S. **Padrões abertos: aplicabilidade em Objetos de Aprendizagem (OAs).** 2009. Disponível em <http://roai.joinville.udesc.br:8080/xmlui/bitstream/handle/123456789/73/SBIE_OA_com_autoria_V14.pdf?sequence=1>. Disponível em: 31 maio 2013.

DROTEK. **Efeito Coriolis.** Disponível em: <<http://www.frigoletto.com.br/GeoFis/coriolis.htm>>. Acesso em: 23 abr. 2013.

ENGENHARIA VISUAL. **Unidades de medida da luz.** 2012. Disponível em: <<http://www.eav.eng.br/tech/fisica/Unidadesdemedidadaluz.html>>. Acesso em: 10 abr. 2013.

EVOBOOKS. Disponível em <<http://www.evobooks.com.br/>>. Acesso em: 31 maio 2013.

FELKER, D.; DOBBS, J. **Android Application Development for Dummies**. 1. ed. Indianapolis: Wiley Publishing, 2011. 384 p.

FUENTES, R. C. **Apostila de Automação Industrial**. Disponível em: <http://w3.ufsm.br/fuentes/index_arquivos/CA03.pdf>. Acesso em: 4 abr. 2013.

GALLO, L. A. **Fotossíntese**. 2011. Disponível em <<http://docentes.esalq.usp.br/luagallo/fotossintese.html>>. Acesso em: 4 ago. 2013.

GONCALVES, J. C. **Uso da plataforma Android em um protótipo de aplicativo coletor de consumo de gás natural**. 2011. Disponível em <http://www2.dainf.ct.utfpr.edu.br/esp/monografias-de-especializacao-da-turma-vi-2010-2011/CT_JAVA_VI_2010_10.PDF/at_download/file>. Acesso em 4 ago. 2013.

GOTARDO, R. A; SOUZA, H. A; JUNIOR, E. H; VIANA, D. B. G. **Teorias de aprendizagens na EAD: fundamentação no uso dos recursos de design instrucional e design interacional**. 2012. Disponível em <<http://sistemas3.sead.ufscar.br/ojs1/index.php/sied/article/viewFile/365/180>>. Acesso em: 15 dez. 2013.

HALL, D. **Fotossíntese**. Vol. 10. Tradução e notas de Antônio Lambertini. São Paulo: E.P.U. Editora Pedagógica e Universitária Ltda, 1980.

IBM. **Introdução ao Desenvolvimento do Android**. Disponível em <<http://www.ibm.com/developerworks/br/library/os-android-devel/>>. Acesso em: 20 nov. 2013.

ICNT REDOXOMA. Instituto de Química – USP. 2013. Disponível em <<http://www2.iq.usp.br/redoxoma/faq.html>>. Acesso em: 4 ago. 2013.

IDC. **Estudo da IDC mostra recorde nas vendas de smartphones no terceiro trimestre de 2013**. 2013. Disponível em <<http://br.idclatin.com/releases/news.aspx?id=1547>>. Acesso em: 15 dez. 2013.

IEEE. **Draft Standard for Learning Object Metadata**. 2002. Disponível em <http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf>. 2002. Acesso em: 31 maio 2013.

INSTITUTO BRASILEIRO DE OPINIÃO PÚBLICA E ESTATÍSTICA. **Número de pessoas com acesso à internet passa de 100 milhões**. 2013. Disponível em <<http://www.ibope.com.br/pt-br/noticias/paginas/numero-de-pessoas-com-acesso-a-internet-passa-de-100-milhoes.aspx>>. Acesso em: 19 dez. 2013.

JÚNIOR, D. B. **Ainda sobre os acelerômetros**. 2008. Disponível em: <http://fisicamoderna.blog.uol.com.br/arch2008-07-06_2008-07-12.html>. Acesso em: 10 abr. 2013.

LAURENCE, J. **Biologia: ensino médio, volume único**. 1.ed. São Paulo: Nova Geração, 2005.

LECHETA, R. R. **Google Android – Aprenda a Criar Aplicações Para Dispositivos Móveis Com o Android Sdk**. 3. ed. São Paulo: Novatec, 2010. 824 p.

LIMA, L. **Pesquisa revela dados sobre a relação de crianças e adolescentes com a tecnologia**. 2013. Disponível em <<http://www.ebc.com.br/infantil/para-pais/2013/01/pesquisa-revela-dados-sobre-a-relacao-de-criancas-e-adolescentes-com-a>>. Acesso em: 10 set. 2013.

LUCENA, S. V. S; DIAS, C. C. L; KEMCZINSKI, A; HOUNSELL, M. S. **Metodologia de desenvolvimento para a construção de Objetos de Aprendizagem**. 2010. Disponível em <<http://pt.scribd.com/doc/37009831/Metodologias-de-Desenvolvimento-para-a-Construcao-de-Objetos-de-Aprendizagem>>. Acesso em: 20 nov. 2013.

MEIER, Reto. **Professional Android 4 Application Development**. 3. ed. Indianapolis: Wrox, 2012. 864 p.

MENEZES, E. F. **Efeito Coriolis**. [200-?]. Disponível em: <<http://www.frigoletto.com.br/GeoFis/coriolis.htm>>. Acesso em: 23 abr. 2013.

MICROSOFT OFFICE. **Adicionar uma tendência ou linha média a um gráfico**. 2013. Disponível em: <<http://office.microsoft.com/pt-br/excel-help/adicionar-uma-tendencia-ou-linha-media-a-um-grafico-HA102809798.aspx>>. Acesso em: 26 abr. 2013.

MILETTE, G.; STROUD, A. **Professional Android Sensor Programming**. 1. ed. Indianapolis: Wrox, 2012. 552 p.

MUSTARO, P. N.; SILVEIRA, I. F.; OMAR, N.; STUMP, S. M. D. 2007. **Structure of Storyboard for Interactive Learning Objects Development”, Learning Objects: and instructional design**. Informing Science Press, p.253-279.

MUZIO, J.; HEINS, T.; MUNDELL, R. **Experiences with Reusable eLearning Objects: From Theory to Practice**. Victoria, Canadá. 2001.

NOMADS, USP. **Sensores**. [200-?]. Disponível em: <<http://www.nomads.usp.br/pesquisas/design/dos/Capacidade/arquivos/sensores.pdf>>. Acesso em: 4 abr. 2013.

OLIVEIRA, L. R. **Desenvolvimento de Objetos de Aprendizagem para dispositivos móveis: iniciação ao m-learning**. 2008. Disponível em <www-app.inf.ufsm.br/bdtg/arquivo.php?id=72&download=1>. Acesso em: 5 jun. 2013.

PEREIRA, L. C. O.; SILVA, M. L. **Android para desenvolvedores**. 2. ed. Rio de Janeiro: Brasport, 2009. 240 p.

PERRENOUD, P. **Dez Novas Competências para Ensinar**. Porto Alegre: Artes Médicas, 2000.

PESSOA, M. C; BENITTI, F. B. V. **Proposta de um Processo para Produção de Objetos de Aprendizagem.** Disponível em < <http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/download/4596/3483>. >. Acesso em: 31 maio 2013.

PRENSKY, M. **Digital Natives, Digital Immigrants.** In: PRENSKY, Marc. [s.l.]: NCB University Press, 2001. Disponível em < <http://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf>>. Acesso em: 20 mar. 2013.

PRESSMAN, R. S. **Engenharia de Software – Uma abordagem profissional.** 7. ed. São Paulo: Artmed, 2011. 770 p.

PROGRAMA DE ENGENHARIA BIOMÉDICA - UFRJ. **Conceito de Sinal.** [200-?]. Disponível em: < <http://www.peb.ufrj.br/cursos/COB781/Ref-02.pdf> >. Acesso em: 4 abr. 2013.

QIEDUCACAO. 2010. **ATP e energia.** Disponível em <<http://www.qieducacao.com/2010/05/atp-e-energia.html>>. Acesso em: 4 ago. 2013.

REVISTA VEJA. **A geração Z - Características e perspectivas de uma juventude que conhece a internet desde a infância.** 2001. Disponível em: <http://veja.abril.com.br/idade/palavra_leitor/geracao_z.html>. Acesso em: 20 mar. 2013.

RIVED. Ministério da Educação. **Conheça o RIVED.** [200-?]. Disponível em <http://rived.mec.gov.br/site_objeto_lis.php>. Acesso em: 25 nov. 2013.

ROSÁRIO, J. M. **Princípios de mecatrônica.** 1. ed. São Paulo: Prentice Hall, 2005. 368 p.

SABER ELETRÔNICA. **10 filtros passa-banda ou passa-faixa.** 2008. Disponível em: < <http://www.sabereletronica.com.br/secoes/leitura/789>>. Acesso em: 26 abr. 2013.

SAKURAY, F. **Conversões digitais analógicos – d/a.** Disponível em: < <http://www2.dc.uel.br/~sakuray/Espec-Comunicacao%20de%20dados/Oscar%20O.%20Filho%20-%20Marcelo%20D.%20Manginelli/pag5.htm>>. Acesso em: 26 abr. 2013.

SANTOS, M.; SCARABOTTO, S. C. A.; MATOS, E. L. M. **Imigrantes e nativos digitais: um dilema ou desafio na educação?** In: Congresso Nacional de Educação – EDUCERE, 10, 2011, Curitiba. Disponível em < http://educere.bruc.com.br/CD2011/pdf/5409_3781.pdf >. Acesso em: 19 dez. 2013.

SHOW DE FÍSICA. **Força magnética s/carga elétrica.** 2008. Disponível em: < http://www.rc.unesp.br/showdefisica/99_Explor_Eletrizacao/paginas%20htmls/For%C3%A7a%20magn%C3%A9tica.htm>. Acesso em: 14 abr. 2013.

SILVA, R.M.G. **Epistemologia e Construção de Materiais Didáticos Digitais**. In: CICILLINI, G.A.; BARAÚNA, S.M. *Formação Docente: saberes e práticas pedagógicas*. Uberlândia: EDUFU, 2006.

SISPESCA. **Declinação Magnética**. Disponível em: <<http://www.sispesca.io.usp.br/outros/cursos/navegacao/sld031.htm>>. Acesso em: 14 abr. 2013.

STRATEGY ANALYTICS. **Strategy Analytics: Android Dominates the Tablet Market in 2013 Q2 with 67 Percent Share of Global Tablet Shipments**. 2013. Disponível em <<http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=5403>>. Acesso em: 15 dez. 2013.

TAIZ, L; ZEIGER, E. **Fisiologia Vegetal**. 4.ed. Tradução de Elaine Romanato Santarém. Porto Alegre: Artmed, 2009.

TAROUCO, L. M. R.; FABRE, M. J. M.; TAMUSIUNAS, F. R. **Reusabilidade de objetos educacionais**. In: RENOTE (Revista Eletrônica de Novas Tecnologias na Educação). Porto Alegre: [s.n.], v.1, n.1, 2003. Disponível em: <<http://www.cinted.ufrgs.br/RENOTE>>. Acesso em: 25 mar. 2013.

TELECO. **Os Semicondutores**. 2013. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialsemicon/pagina_3.asp>. Acesso em: 14 abr. 2013.

VIANA, R. L. **Ímãs permanentes**. Disponível em: <http://fisica.ufpr.br/viana/fisicab/aulas_2009/aula_31.pdf>. Acesso em: 23 abr. 2013.

VIEIRA, C. E. M.; NICOLEIT, L. L. G. **Objeto de Aprendizagem para Suporte à Aprendizagem de Funções**. 2008. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/viewFile/3889/2956>>. Acesso em: 5 nov. 2013.

VIEIRA, C. E.; NICOLEIT, E. R. **Desenvolvimento de Objeto de Aprendizagem, baseado em Especificações de Normatização SCORM, para o Caso de Suporte à Aprendizagem de Funções**. 2007. Disponível em: <<http://www.cinted.ufrgs.br/ciclo9/artigos/4eCarlos.pdf>>. Acesso em: 25 nov. 2013.

VITORINO, J. C. **Filtros de sinais**. Disponível em: <http://www.faccamp.br/apoio/JoseCarlosVotorino/princ_com/AulassobreFiltrosdesinais.pdf>. Acesso em: 23 abr. 2013.

WAECHTER, H. N.; LIMA, M. A. **Desenvolvimento de um objeto de aprendizagem interativo para tablet**. 2013. Disponível em <http://66.7.199.78/~andreenr/Anais-CONAHPA-2013/assets/desenvolvimento_objeto_marcos.pdf>. Acesso em: 31 maio 2013.

WILEY, D.A. *Connecting learning objects to instructional theory: a definition, a metaphor and a taxonomy*. 2001. Disponível em <<http://reusability.org./read/chapters/wiley.doc>> Acesso em: 10 abr. 2013.

WILGES, B. **Um agente pedagógico animado no papel de LMS manipulando objetos inteligentes de aprendizagem**. 2006. Disponível em <http://www2.ufpel.edu.br/prg/sisbi/bibct/acervo/info/2006/mono_beatriz_wilges.pdf>. Acesso em: 10 abr. 2013.

ZANETTE, E. N.; NICOLEIT, E. R.; GIACOMAZZO, G. F. **A produção do material didático no contexto cooperativo e colaborativo da disciplina de Cálculo Diferencial e Integral I, na modalidade de educação a distância, na graduação**. In: VII CICLO DE PALESTRAS SOBRE NOVAS TECNOLOGIAS NA EDUCAÇÃO, 9, 2006, Porto Alegre.

APÊNDICES

APÊNDICE A - Diagrama de classes do projeto

